Wondershare PDFelement

جامعة الحسن الأول
UNIVERSITÉ HASSAN 1er

كلية العلوم و التقنيات سطات
FACULTÉ DES SCIENCES ET TECHNIQUES SETTAT

Les Base de données décisionnelles

Implémentation des Data Warehouse et génération des Hypercubes dans **ORACLE**

Table des matières

I- Bases de données relationnelles : 2

II- Les systèmes d'information décisionnels 2

III- Base de données Décisionnelle (Data Warehouse) 3

III-1 Définition 3

III-2 Architecture d'un DW 4

IV- Les BDR et les DW 4

V- Modélisation d'un DW 6

V-1 Schéma étoile (Star Schema) 7

V-2 Schéma en flocon (Snowflake Schema) 7

VI- Implémentation d'un datawarehouse dans Oracle 9

IV-1- Les DWs ROLAP 12

IV-2- Les DWs MOLAP 12

IV-3- Les DWs HOLAP 12

VII- Outils Oracle pour générer les Cubes de données dans un DW 13

VII-1- mappage des objets de DW aux objets des hypercubes 13

a- Les dimensions 14

b- Les niveaux 15

c- Les hiérarchies 16

d- Les Attributs 16

e- Les cubes 16

f- Les mesures 17

VII-2-Génération des Hypercubes dans Oracle à l'aide de l'option OLAP 18

VII-1-1 Création des dimensions 19

a- Syntaxe SQL de Création des Dimensions 19

b- Création des dimensions avec SQL-developper (Option OLAP) 21

Wondershare PDFelement

Wondershare PDFelement

VII-1-2 Création des cubes de données : 27
a- Syntaxe SQL de création des cubes de données 27
b- Création des cubes de données avec l'Option OLAP 27
VII-3 Exemple d'Implémentation de CUBES dans Oracle à l'aide de l'SQL 32
a- Implémentation de l'entrepôt de données 33
b- Construction des hypercube de Données 34
VII-2 Opération sur les hypercubes de données 35
VII-2-1 Opération de sélection (Slicing et Dicing) 36
VII-2-2 Opérations de changement de granularité (Roll-up et Drill-Down) 37
VII-3 Les fonction d'agrégation dans les DWs 38
VII-3-1 ROLLUP 38
V-4-1 CUBE 39
V-4-1 GROUPING SETS 39
V-4-1 Fonction Decode 40

Wondershare PDFelement

I- Bases de données relationnelles :

En 1970, au moment où les systèmes reposant sur le modèle hiérarchique étaient en plein développement, Edgar Frank Codd publia un article où il proposait de stocker des données hétérogènes dans des tables, permettant d'établir des relations entre elles. De nos jours, ce modèle est extrêmement répandu. Un premier prototype de Système de gestion de bases de données relationnelles (SGBDR) est construit dans les laboratoires d'IBM. Depuis les années 1980, cette technologie a mûri et a été adoptée par l'industrie. En 1987, le langage SQL, une extension de l'algèbre relationnelle, est standardisé. À l'heure actuelle, les SGBDR sont présents dans de nombreux logiciels, sont très répandus dans les bases de données et représentent une industrie très développée.

Les applications interrogent les données d'une BDDR via la commande SELECT du langage SQL (select) et les mettent à jour via les commandes Update, Delete et Insert. Ces derniers constituent les transactions qui doivent avoir les propriétés ACID (Atomicité, Cohérence, Isolation et Durabilité). Ce type d'opérations est appelé On_line Transaction Processing (OLTP).

II- Les systèmes d'information décisionnels

Les systèmes d'information décisionnel diffusent des informations fortement synthétisées nécessaires à la prise de décision. Ces informations sont représentées sous forme de rapport, de

Wondershare PDFelement

tableaux de bord, de graphiques, de synthèses, d'agrégat, résumé, etc….parmi ces système on site le système d'aide à la décision (SIAD).

Contrairement aux application OLTP qui consultent et mettent à jour les données des BD opérationnelle, les SIAD lisent seulement les données pour avoir des informations synthétisées.

Cependant il y a beaucoup de problèmes lorsque le SIAD et les applications OLTP partagent les mêmes données :

- Un SIAD a besoin de faire des analyses sur des données historiques. Ce qui est impossible dans les BDD de production à cause des transactions qui changent les données.
- L'utilisation des BDD en mode multi-utilisateur implique des opérations de verrouillage des données et donc des problèmes de performance car les requêtes analytiques, réalisées par les SIAD, demandent l'accès à un très grand nombre de tuples de différentes relations.
- Le modèle relationnel qui se base sur la non redondance et la normalisation a montré ses limites dans le traitement complexe d'analyses des données. Incapacité donc de fournir des services décisionnels à cause du coût élevé des calculs dans les jointures. Par conséquent il sera dans l'impossible de créer des Requêtes complexes. On ajoute aussi l'incapacité de navigation sur les données pour les analyses.

La solution est de séparer la BDD orienté Transaction de la BDD orienté Aide à la décision. D'où la naissance des entrepôts de données (ou Dataware House)

Wondershare PDFelement

III- Base de données Décisionnelle (Data Warehouse)

III-1 Définition

Un DW est une collection de données orientées sujets, intégrées, non volatiles et historisées, organisées afin de servir de source de données à des applications décisionnelles. C'est un gigantesque tas de données fortement agrégées à partir d'un ensemble de données provenant de plusieurs sources, principalement de BDD transactionnelle (BDD de production)

L'objectif d'un DW est de faciliter des requêtes sur de grands ensembles des données, la plupart du temps sous forme d'agrégats (GROUP BY) afin d'en obtenir une vision synthétique (propre à la prise de décision).

Les Data Warehouse représentent le concept central d'un SI décisionnel. Ils représentent une vision centralisée et universelle de toutes les informations de l'entreprise. C'est une structure qui a pour but de regrouper les données de l'entreprise pour des fins analytiques et pour aider à la décision stratégique. Cette dernière étant une action entreprise par les décideurs de l'entreprise et qui vise à améliorer, quantitativement ou qualitativement, la performance de l'entreprise.

Wondershare PDFelement

Le data Warehouse n'est donc pas un produit ni un logiciel qu'on va se procurer mais un environnement qui se bâtit.

III-2 Architecture d'un DW

La construction d'un entrepôt de données comporte deux phases : la phase d'intégration, et la phase d'analyse.
Dans la phase d'intégration, les données sont collectées de plusieurs sources hétérogènes pour être nettoyées et intégrées afin de construire l'entrepôt.
Dans la phase d'analyse, l'entrepôt conçu est exploité pour répondre aux requêtes des décideurs. Ces requêtes permettent d'effectuer des analyses sur les données afin de permettre la prise de décision (figure 1).

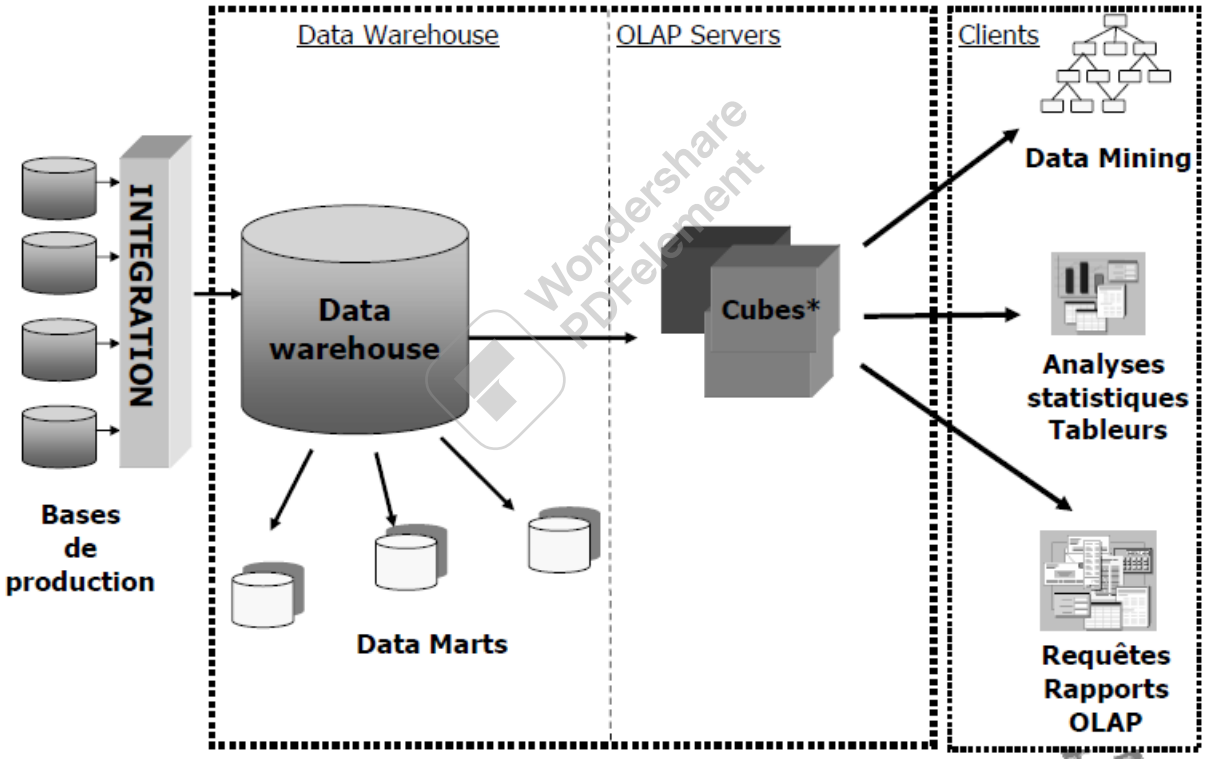


Figure 1 : Architecture d'un entrepôt de données

IV- Les BDR et les DW

Dans Les BDR le mode de travail est transactionnel. Les requêtes sont de type OLTP (ON-LINE Transaction Processing), elles permettent d'insérer, modifier, interroger des Informations rapidement, efficacement et en sécurité. Ces transactions doivent respecter certaine propriété ACID qui sont :
L'Atomicité : L'atomicité est la garantie qu'une série d'opérations de base de données dans une transaction atomique se produiront toutes (opération réussie) ou qu'aucune ne se produira

Wondershare PDFelement

(opération infructueuse). Chaque transaction est traitée comme une seule "unité", qui réussit complètement ou échoue complètement :

La cohérence : garantit qu'une transaction ne peut faire passer la base de données que d'un état cohérent à un autre cohérents, en préservant les invariants de la base de données comme les contraintes d'intégrité et les déclencheurs.

L'Isolement : garantit que l'exécution simultanée des transactions laisse la base de données dans le même état que celui qui aurait été obtenu si les transactions avaient été exécutées de manière séquentielle. C'est l'objectif principal du contrôle de la concurrence

La Durabilité : garantit qu'une fois qu'une transaction a été validée, elle le restera même en cas de défaillance du système (une panne de courant ou un crash). Cela signifie que les transactions terminées sont enregistrées dans une mémoire non volatile .

Dans Les DW le mode de travail est Décisionnel. Les requêtes sont de type OLAP (ON-LINE Analytics Processing), elles permettent de regrouper, organiser des informations provenant de sources diverses, les intégrer et les stocker pour donner à l'utilisateur une vue orientée métier. Ceci pour retrouver et analyser l'information facilement et rapidement.

Dans les deux modes le SQL reste le langage des requêtes avec la possibilité d'agréger les données à l'aide de la commande (Group By). Ces types de requête sont très couteuses dans le mode transactionnel. Ceci est dû au nombre important de jointure qui font augmenter le nombre d'opération algébrique en entre les tables. Ce n'est pas le cas dans le mode décisionnel où le nombre de jointure est très réduit.

Wondershare PDFelement

Dans le tableau-1 nous présentons une comparaison entre une BDD classique et un entrepôt de données.

	OLTP	**OLAP**
Modèle	Entité association	(Fait dimension). Schéma en étoile, flocon de neige ou en constellation
Application	Production	Aide à la décision
Donnes	Normalisées, non agrégées	Dé normalisées, agrégées
Requête	Simples, nombreuses, Régulières, prévisibles, Répétitives	Complexes, peu Nombreuses, irrégulières, Non prévisibles
Nombre moyen de tuples invoqués par requête	Dizaines	Millions
Volume	Des Giga Octets	Des Téra Octets
Ancienneté des données	Récente, mises à jour	Historique

Pas les mêmes objectifs : en effet les BDR sont développées pour des fins de production par contre les DW sont construites pour des fins d'analyse.

Wondershare PDFelement

Pas les mêmes données : par ce que les données nécessaires à l'analyse sont multidimensionnelles (ventes par produit par ville et par année), tandis que les tables d'une BDR représentent une vue aplatie.

Pas les mêmes traitements : la modélisation relationnelle ne permet de faire des traitements complexes et en plus pour réaliser ces traitements l'utilisateur doit savoir comment trouver les liens entre les tables pour créer des vues multidimensionnelles.

Il est donc nécessaire de disposer d'une structure de stockage adaptée à l'OLAP, i.e. permettant d'une part de représenter les données dans plusieurs dimensions, et d'autre part manipuler les données facilement et efficacement.

V- Modélisation d'un DW

La modélisation des données est un élément essentiel de l´informatique décisionnelle. Elle s´appuie sur des concepts permettant d´optimiser la restitution des données, selon les différents axes métiers de l´entreprise. Cette méthode de modélisation part du processus métier. Elle s'appuie sur le couple *Fait/Dimension*

1. Identification du processus métier analysé.
2. Établissement du niveau de granularité de l'analyse
3. Établissement des dimensions qui s'appliquent à chaque ligne de fait
4. Identification des caractéristiques numériques qui s'appliquent à chaque ligne de fait.

Wondershare PDFelement

Cette méthode demande une vision complète et décrite du processus métier et de son implémentation. Elle permet de construire le tableau qui croise les processus métier (et les différentes étapes des processus métier) et les dimensions :

Dimension / Processus	Date	Produit	Magasin	Entrepôt	Promotion
Vente	×		×		×
Stock Entrepôt	×	×		×	
Commande	×	×			×

Il existe deux schémas principaux possibles pour la modélisation d'un DW: le *schéma en étoile* et le schéma en flocon. Quel que soit le modèle considéré, on distinguera la table des **faits** qui contient l'information à analyser (par exemple les ventes) des tables de **dimensions** qui contiennent les informations sur les dimensions d'analyse (par exemple le lieu, le temps, la description du produit).

Wondershare PDFelement

Deux types d'information qui peuvent caractériser un fait. Les mesures numériques propres à chaque instance de faits (quantité, prix, …) et les dimensions caractéristiques plus ou moins complexes (fournisseur, produit, …).

Les mesures numériques propres à chaque fait sont l'objet de calculs d'analyse (montant global, moyenne globale, …) et les dimensions constituent des facteurs de sélection des faits étudiés (produits de telle ou telle marque, …).

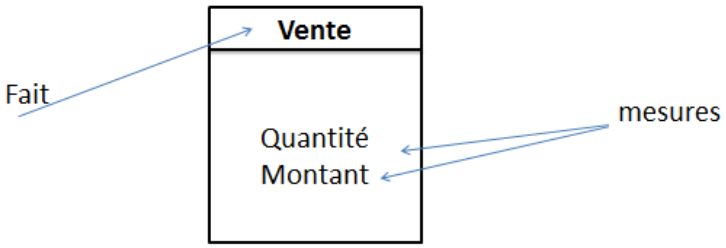


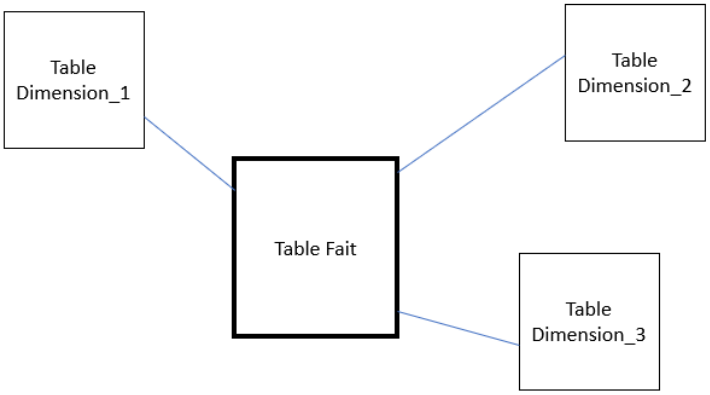
Figure-1

V-1 Schéma étoile (Star Schema)

Ce type de schéma est décrit sur la figure 2. La table des faits contient les clés étrangères vers les tables de dimension. L'ensemble de ces clés étrangères forment la clé primaire de la table des faits. Les tables dimensions ne sont pas normalisées. Un exemple est donné dans la figure3.

V-2 Schéma en flocon (Snowflake Schema)

Dans ce modèle, les tables de dimensions sont normalisées pour constituer une structure hiérarchique (figure-4). Ceci a pour avantage de faciliter la maintenance des tables de dimension et de réduire la redondance, mais d'un autre coté le nombre de jointure augmente ce qui va rendre les calculs couteux. Dans l'exemple de la figure5 le schéma en flocon est généré à partir du schéma de la base de données transactionnel présentée dans la figure-5.



Wondershare PDFelement

Figure-2 : Schéma en étoile

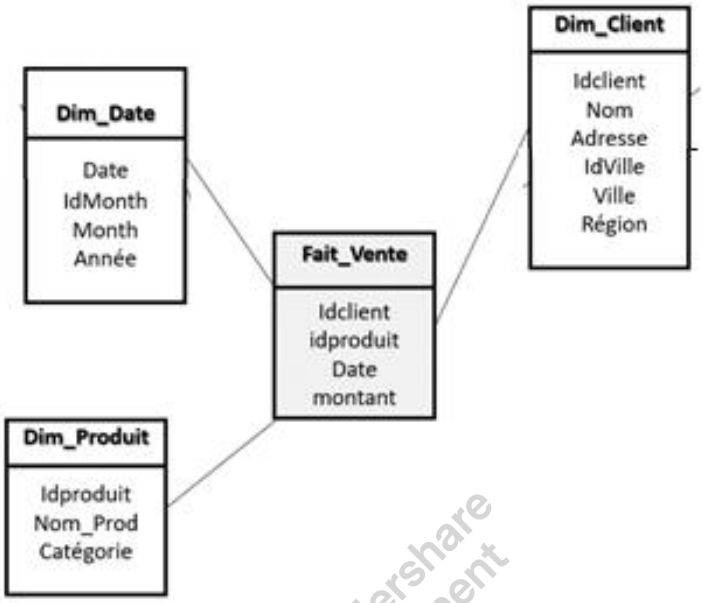


Figure-3 : Exemple de Schéma en étoile d'une BDD décisionnelle

Wondershare PDFelement

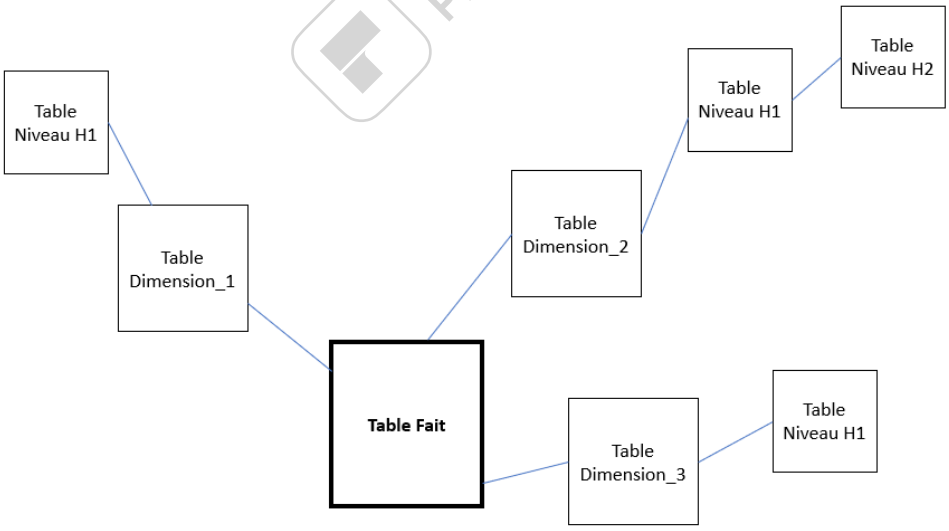


Figure-4: Schéma en flocon

Wondershare PDFelement

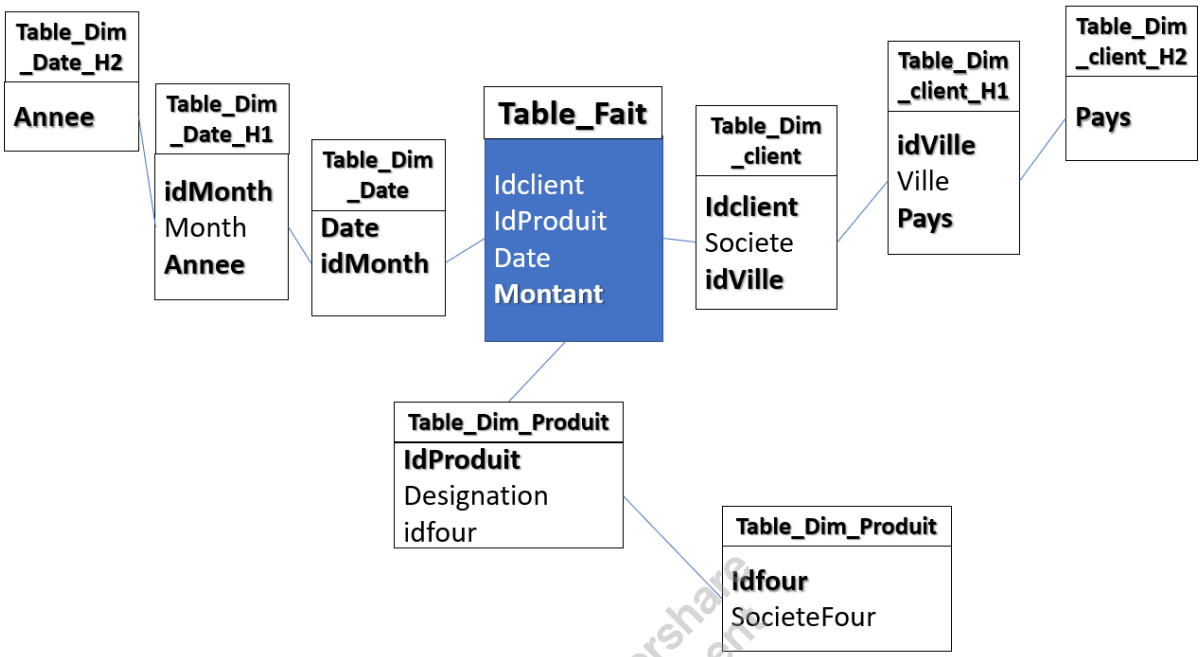


Figure-5: Schéma en flocon d'une BDD décisionnelle avec une normalisation des tables dimension.

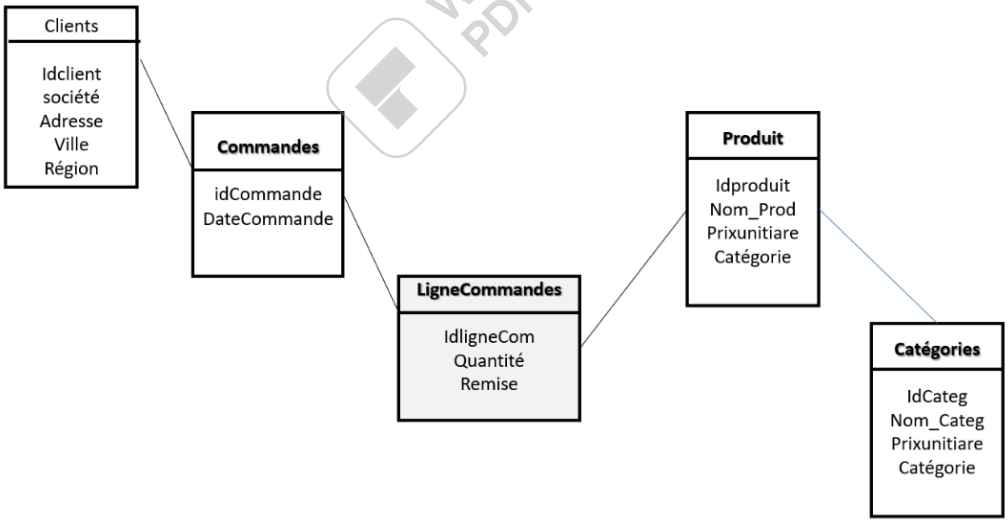


Figure-6 : Schéma d'une base de données transactionnel.

VI- Implémentation d'un datawarehouse dans Oracle

L'utilisation des vue matérialisés Oracle reste un moyen efficace pour implémenter un DW qui sera alimenté par la BDD transactionnelle de production. Les options de rafraichissement permettent de définir si ce rafraichissement se fait ou non automatiquement, et à quels moments il doit s'exécuter. La syntaxe de la création des MVs est :

Pour la table Dimension:

Wondershare PDFelement


```
CREATE MATERIALIZED VIEW Dim_TABLE
REFRESH [FAST | COMPLETE | FORCE ]
ON [COMMIT | DEMAND ]
[[ENABLE | DISABLE] QUERY REWRITE]
AS
SELECT Att1, Att2, …..
FROM .Table.. ;
```

Pour la table Fait :

```
CREATE MATERIALIZED VIEW Fait
REFRESH [FAST | COMPLETE | FORCE ]
ON [COMMIT | DEMAND ]
[[ENABLE | DISABLE] QUERY REWRITE]
AS
SELECT ... , AGGREGATION()
FROM ...
WHERE ...
GROUP BY ... ;
```

Remarque :
Dans le cas d'un schéma en étoile (dimension dénormalisées) il faut introduire pendant la création des dimension un identifiant pour chaque niveau d'agrégation.

Wondershare PDFelement

Les types d'actualisation suivants sont disponibles.

- **FAST :** Un rafraîchissement rapide est tenté. Si les journaux de vue matérialisée ne sont pas présents sur les tables source à l'avance, la création échoue.
- **COMPLET :** Le segment de table supportant la vue matérialisée est tronqué et repeuplé intégralement à l'aide de la requête associée.
- **FORCE :** Un rafraîchissement rapide est tenté. Si cela n'est pas possible, un rafraîchissement complet est effectué.

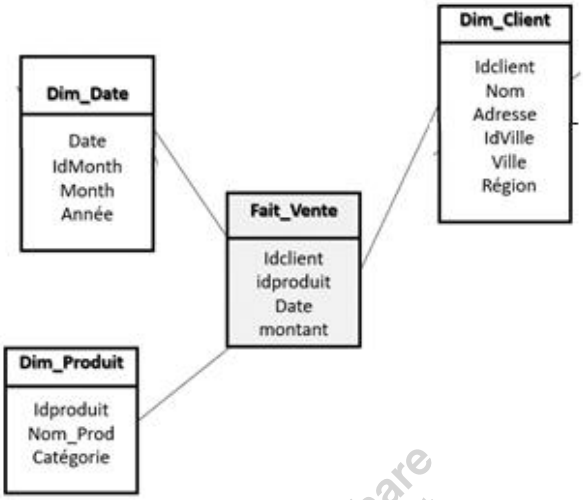
Une actualisation peut être déclenchée de deux manières.

- **ON COMMIT :** Le rafraîchissement est déclenché par une modification de données validée dans l'une des tables dépendantes.
- **A LA DEMANDE :** Le rafraîchissement est initié par une requête manuelle ou une tâche planifiée.

QUERY REWRITE clause : indique à l'optimiseur si la vue matérialisée doit être prise en compte pour les opérations de réécriture de requête. Dans le cas où une requête semblable à celle utilisée dans la VM, la réécriture de la requête est déclenchée pour tirer parti des données pré-agrégées dans la vue matérialisée, au lieu que la session fasse le travail pour elle-même.

Wondershare PDFelement

Exemple : on obtient le modèle du datawehouse ci-dessous à partir des vues matérialisées appliquée par le modèle transactionnel (figure-6)



Wondershare PDFelement

Creation des table dimension:

```
create materialized view Table_Dim_Clients as
select idclient, nom, adresse,
ville||'_'||region||'_'||pays as idville, ville, region
from clients

create MATERIALIZED VIEW table_Dim_Dates as
select distinct extract(year from datecommande) as ANN,
extract(month from datecommande)||'_'||extract(year from datecommande) as idmois,
extract(month from datecommande) as mois,
datecommande as date
from commandes;

create MATERIALIZED VIEW table_Dim_Produit as
select distinct idproduit, Nom_prod, categorie,
from produits;
```

Creation de la table fait:

```
Create materialised View Fait
Refresh COMLETE on DEMAND AS
select idclient, produits.idproduit, datecommande,
sum(prixunitaire*quantite*(1-remise/100)) as CA
from commandes, lignecommandes, produits
where commandes.idcommande=lignecommandes.idcommande
```

Wondershare PDFelement

AND lignecommandes.idproduit=produits.idproduit
group by idclient, produits.idproduit, datecommande

IV-1- Les DWs ROLAP

On nomme ROLAP l'approche Relationnel OLAP. Les données sont stockées sous la forme de tables relationnelles. Elles sont modélisées sous la forme de schémas en étoile ou en flocon. Les requêtes multidimensionnelles doivent alors être traduites en requêtes relationnelles (SQL).
Avantages :
Ce modèle est avantageux en termes de capacité de stockage, Peut gérer de grandes quantités de données et peut tirer parti des différentes fonctionnalités déjà existantes dans les bases de données relationnelles.
Inconvénients
Les requêtes sont difficiles à définir et à mettre en œuvre et sont coûteuse. Chaque rapport ROLAP étant essentiellement une requête SQL (ou plusieurs requêtes SQL) dans la base de données relationnelle, le temps de requête peut être long si la taille des données sous-jacentes est importante.

Wondershare PDFelement

IV-2- Les DWs MOLAP

On nomme MOLAP l'approche Multidimensionnelle OLAP. La technologie de stockage est multidimensionnelle. Les données sont stockées sous la forme de tableaux multidimensionnels (figure-5bis) (hypercube), des index multidimensionnels sont définis. Les données dans ce type de stockage seront plus compressées par rapport à la solution ROLAP. Toutefois, ce type d'implémentation de DW demande des SGBD multidimensionnels avec un langage de manipulation des données qui n'est pas aussi développé que le SQL.
Avantages
Les Performances sont plus parfaites puisque les cubes MOLAP sont conçus pour une récupération rapide des données, et optimales pour les opérations de découpage. MOLAP peut donc effectuer des calculs complexes : tous les calculs ont été pré-générés lors de la création du cube. Par conséquent, les calculs complexes sont non seulement réalisables, mais ils reviennent rapidement.
Inconvénients
La quantité de données à gérer est limitée : Tous les calculs étant effectués lors de la création du cube, il est impossible d'inclure une grande quantité de données dans le cube lui-même. En plus les SGBD sont propriétaire.

IV-3- Les DWs HOLAP

On nomme HOLAP l'approche Hybride OLAP. Cette technologie combine les deux solutions précédentes. Les données détaillées sont stockées dans une base de données relationnelle, et les données agrégées dans une base multidimensionnelle.
Ci-dessous on trouve des SGBD qui permettent de gérer et de manipuler des DWs ROLAP et MOLAP

Wondershare PDFelement

Produit	Editeur	Type
Essbase	Arbor Software	MOLAP
DB2 OLAP Server	IBM	ROLAP/MOLAP
Metacube	Informix	ROLAP
SQL Server (2000)	Microsoft	ROLAP
Express Server	Oracle	MOLAP
9i OLAP	Oracle	ROLAP/MOLAP

VII- Outils Oracle pour générer les Cubes de données dans un DW

Avec la version Oracle 9i, Oracle a décidé d'intégrer les fonctionnalités OLAP au sein du noyau du SGBD. Des fonctionnalités liées aux entrepôts de données comme les vues matérialisées sont présentes. Ces derniers permettent de stocker de gros volumes de données modélisés sous la forme d'un schéma en étoile ou flocon). L'accès aux données multidimensionnelles est possible via SQL à travers des vues, via le langage OLAP DML, ou via la Java OLAP API.

Le modèle multidimensionnel sur lequel s'appuie l'analyse OLAP permet de définir des hypercubes (appelé aussi des cubes de données). A partir d'un schéma en étoile ou en flocon on peut construire un hypercube de donnée de n dimension.

Wondershare PDFelement

- Les tables de faits correspondent à des **cubes**.
- Les colonnes de données des tables de faits correspondent à des **mesures**.
- Les contraintes de clé étrangère dans les tables de faits identifient les tables de dimension.
- Les tableaux de dimensions identifient les **dimensions**.
- Les clés primaires dans les tables de dimension identifient les membres de dimension de **niveau (Level) de base**.
- Les colonnes parentes des tables de dimension identifient les membres de dimension de **niveau (Level) supérieur**.
- Les colonnes des tables de dimension contenant des descriptions et des caractéristiques des membres de dimension identifient les **attributs**.

VII-1- mappage des objets de DW aux objets des hypercubes

Les objets de données de l'entrepôt de données sont donc représentés par les objets relationnels suivants :

- Tables ou vues de faits
- Tables ou vues de dimension basées sur les niveaux

Oracle OLAP permet de mapper le schéma de l'entrepôt de données aux objets de données multidimensionnels. Ces derniers sont :

- Dimensions
- Niveaux
- Hiérarchies
- Attributs de niveau

Wondershare PDFelement

- Attributs de dimension
- Cubes
- Les mesures
- Mesurer les dossiers

a- Les dimensions

Ils identifient et catégorisent vos données. Les membres de dimension sont stockés dans une table de dimension. Chaque colonne représente un niveau particulier dans une hiérarchie. Dans un schéma en étoile, les colonnes sont toutes dans la même table et dans un schéma de flocon de neige, les colonnes sont dans des tables séparées pour chaque niveau.
Dans Oracle version Enterprise Manager, la définition d'une dimension dans l'entrepôt de données crée un objet de dimension de base de données. Un objet dimension contient les détails de la relation parent-enfant entre les colonnes d'une table de dimension.

Dimension temps : C'est une dimension importante dans les traitements multidimensionnels. Une dimension de temps a des attributs spéciaux qui prennent en charge les périodes régulières et irrégulières.

- *Les périodes régulières* : telles que les semaines, les mois et les années, sont bien définies sur les calendriers standard.
- *Les périodes irrégulières* : telles que les calendriers promotionnels et les périodes saisonnières, ne sont pas évidentes sur les calendriers standard. Ils se chevauchent souvent (même dans la mesure où une période de temps est un sous-ensemble d'une autre période de temps) ou ont des écarts entre eux.

Wondershare PDFelement

Le tableau des dimensions de temps doit contenir les colonnes suivantes pour fournir une assistance à temps plein :

- Valeurs pour tous les membres de dimension, avec une colonne pour chaque niveau de récapitulation (comme les semaines, les trimestres et les années).
- Des identifiants pour chaque niveau permettant de garantir l'unicité d'une valeur du membre d'un niveau, tels que ID_month='2010_02' qui va identifier le mois de février de l'année 2010. ID_QUARTER='QUA_1_2010' qui va identifier le 1er trimestre de 2010.
- Un attribut date de fin pour chaque niveau, tels que WEEK_ENDDATE, QUARTER_ENDDATE et YEAR_ENDDATE. Ces colonnes doivent avoir un DATE type de données. Leurs valeurs identifient le dernier jour de la période.
- Un attribut temps pour chaque niveau, tels que WEEK_TIMESPAN, QUARTER_TIMESPAN, et YEAR_TIMESPAN. Ces colonnes doivent avoir

Wondershare PDFelement

un NUMBER type de données. Leurs valeurs identifient le nombre de jours dans la période.

Il faut noter que La fonction de gestion OLAP d'Oracle Enterprise Manager prend en charge la création et le remplissage de tables de dimension de temps avec ces colonnes.

Exemple1 : Dimension temps dans un schéma en étoile

Colonne	Valeur	Type de données	Commentaire
ID_YEAR	YR2010	Varchar2	Niveau 3
YEAR_DESC	2010	Varchar2 ou Number	Attribut
YEAR_ENDDATE	31/12/2010	Date	Attribut
YEAR_TIMESPAN	366	Number	Attribut
ID_QUARTER	QUA_1_2010	Varchar2	Niveau 2
QUARTER_DESC	1	Varchar2 ou Number	Attribut
QUARTER _ENDDATE	31/03/2010	Date	Attribut
QUARTER _TIMESPAN	91	Number	Attribut
ID_MONTH	MO_3_2010	Varchar2	Niveau 1
MONTH_DESC	3	Varchar2 ou Number	Attribut
MONTH _ENDDATE	31/03/2010	Date	Attribut
MONTH _TIMESPAN	31	Number	Attribut

Wondershare PDFelement

Exemple2 : Dimension temps dans un schéma de flocon de neige

Dans ce cas les niveaux seront structurés dans des tables différentes. Dans l'exemple précèdent on aura trois tables. Une table pour le niveau « YEAR » et une deuxième table pour le niveau « QUARTER » et une troisième table pour le niveau le plus bas « MONTH ».

b- Les niveaux

Les niveaux regroupent les données pour l'agrégation et sont utilisés en interne pour le calcul. Chaque niveau représente une position dans la hiérarchie.

Wondershare PDFelement

c- Les hiérarchies

Une **hiérarchie** est un moyen d'organiser les données selon des niveaux. Chaque dimension est formée par un ensemble de niveaux qui sont à leur tour structurées de manière hiérarchique. Cette structuration permet de manipuler d'une manière efficace les agrégations pour l'analyse et l'affichage. Les hiérarchies de dimensions permettent aux utilisateurs de reconnaître les tendances à un niveau d'agrégation, de descendre aux niveaux inférieurs pour identifier les raisons de ces tendances et de remonter aux niveaux supérieurs pour voir l'impact de ces tendances sur un secteur plus large de l'entreprise. Dans une hiérarchie, chaque niveau représente le total agrégé des niveaux inférieurs.

Les membres d'une hiérarchie à différents niveaux ont une relation *parent-enfant* un-à-plusieurs. Par exemple, QUA1 et QUA2 sont les enfants de year_2001, ainsi year_2001 est le parent de qtr1 et qtr2.

d- Les Attributs

Les attributs fournissent des informations descriptives sur les données et sont généralement utilisés pour l'affichage. On distinct :

Attributs de niveau

Les attributs de niveau fournissent des informations supplémentaires sur les membres de dimension à un niveau particulier d'une hiérarchie de dimension.
Exemple : vous pouvez avoir des colonnes pour le numéro d'employé (ENUM), le nom de famille (LAST_NAME), le prénom (FIRST_NAME) et le poste téléphonique (TELNO). ENUM est une colonne de niveau, car il s'agit d'une colonne clé. LAST_NAME, FIRST_NAME et TELNOsont des attributs. Même s'ils sont dimensionnés par ENUM, ils ne prennent pas de mesures appropriées car il s'agit de textes descriptifs plutôt que de mesures commerciales.

Attributs de dimension

Les attributs de dimension spécifient des regroupements d'attributs de niveau pour une dimension spécifique. Un exemple d'attribut de dimension est « end date », qui est requis pour les dimensions de temps. Si une dimension de temps à des niveaux de mois, de trimestre et d'année, end date identifie la dernière date de chaque mois, de chaque trimestre et de chaque année. Au sein d'un schéma relationnel, les trois attributs de niveau qui composent l'end date attribut de dimension seraient stockées dans des colonnes avec des noms comme month_end_date, quarter_end_date et year_end_date.

e- Les cubes

Les cubes sont les objets qui associent les mesures à leurs dimensions. Toutes les mesures associées à un cube ont exactement la même dimensionnalité.

Wondershare PDFelement

Wondershare PDFelement

Les arêtes d'un cube sont définies par ses dimensions. Bien qu'il n'y ait pas de limite au nombre d'arêtes sur un cube, les données sont souvent organisées à des fins d'affichage le long de trois arêtes, appelées bord de ligne, bord de colonne et bord de page. Une seule dimension ou plusieurs dimensions peuvent être placées sur une arête. Par exemple, les données de ventes peuvent être affichées avec le produit et le canal sur le bord de la ligne, l'heure sur le bord de la colonne et le client sur le bord de la page.

f- Les mesures

Les mesures peuvent être organisées dans des **dossiers de mesures,** ce qui facilite la navigation des données par domaine d'activité. Les dossiers de mesure sont également appelés **catalogues**.

Alors que les dimensions et les mesures sont associées aux schémas qui contiennent leurs données source, les dossiers de mesures sont indépendants du schéma. Chaque client OLAP peut afficher tous les dossiers de mesures définis dans l'instance Oracle.

Puisque les mesures sont généralement multidimensionnelles et pour que cette dernière soit significative, un membre de chaque dimension doit qualifier une valeur unique d'une mesure.
Les dimensions peuvent être munies de hiérarchies. On distingue alors les hiérarchies simples des hiérarchies multiples (figure-6). Le niveau ALL correspond à l'agrégation totale.
Le contenu des cellules de l'hypercube désigne une mesure. Cette dernière peut être additive, semi-additive ou non additive :
Mesure non additive : la somme des valeurs des cellules ne donne aucun sens. Exemple : si la cellule contient la moyenne des ventes.
Mesure semi additive : la somme des mesures ne peut se faire que sur certaines dimensions. Exemple : la quantité vendue par produit, par mois mais non pas par client.
Mesure additive : la somme des mesures peut se faire sur toutes les axes d'analyses et suivant le niveau hiérarchique demandé.

Wondershare PDFelement

Wondershare PDFelement

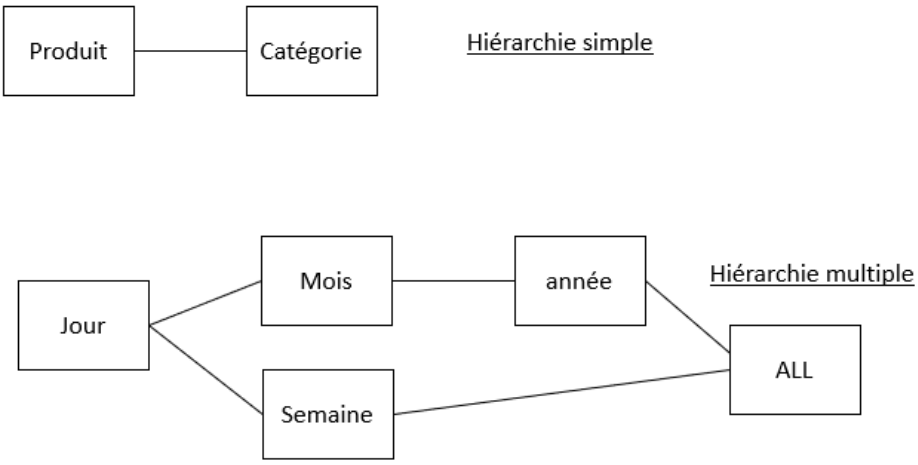


Figure-6 : hiérarchisation des dimensions

VII-2-Génération des Hypercubes dans Oracle à l'aide de l'option OLAP

Soit le modèle en flocon ci-dessous :

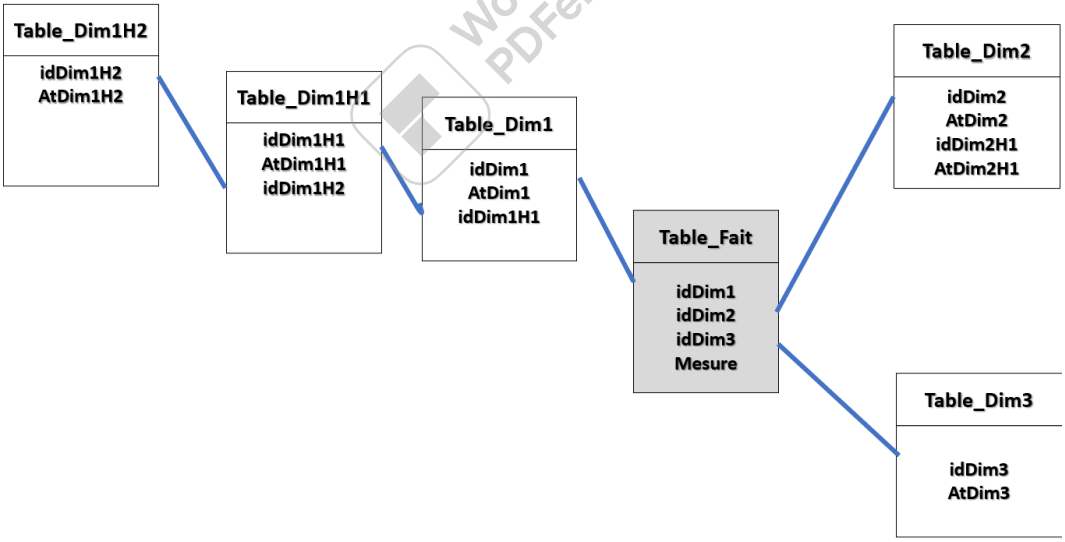


Figure-7 : schéma en flocon

On veut implémenter dans oracle le schéma multidimensionnel suivante

Wondershare PDFelement

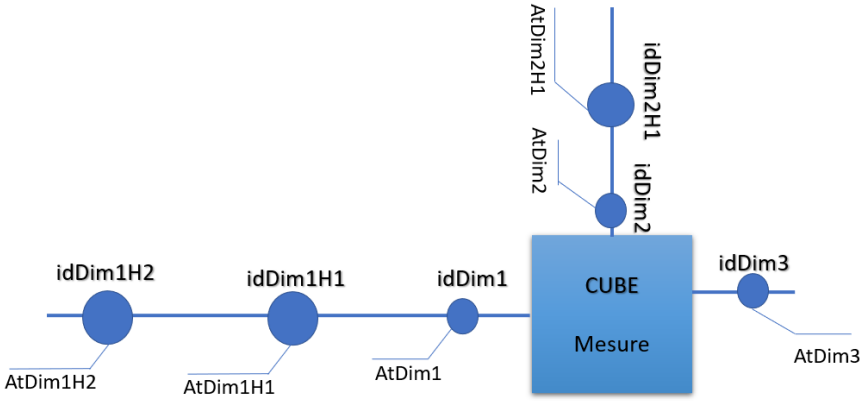


Figure-8 : schéma multidimensionnel

VII-1-1 Création des dimensions

a- Syntaxe SQL de Création des Dimensions

Puisque la table_Dim1 est normalisée les colonnes source des niveaux ne proviennent pas de la même table. Par conséquent il faut préciser dans la création de la dimension les jointures entre les tables des niveaux. Ceci sera pris en charge par la clause JOIN KEY.

```
Create DIMENSION Dim1
LEVEL idDim1 IS (Table_Dim1.idDim1)
LEVEL idDim1H1 IS (Table_Dim1H1.idDim1H1)
LEVEL idDim1H2 IS (Table_Dim1H2.idDim1H2)
HIERARCHY H1Dim1 (
idDim1 CHILD OF
idDim1H1 CHILD OF
idDim1H2
JOIN KEY Table_Dim1.idDim1H1 REFERENCES idDim1H1
JOIN KEY Table_Dim1H1.idDim1H2 REFERENCES idDim1H2)
ATTRIBUTE idDim1 DETERMINES (AtDim1)
ATTRIBUTE idDim1H1 DETERMINES (AtDim1H1)
ATTRIBUTE idDim1H2 DETERMINES (AtDim1H2);
```

Puisque la table_Dim2 est dénormalisée donc les colonnes source des niveaux proviennent de la même table.

Wondershare PDFelement


```
create DIMENSION Dim2
LEVEL idDim2 IS (Table_Dim2.idDim2)
LEVEL idDim2H1 IS (Table_Dim2.idDim2H1)
HIERARCHY H1Dim2 (
idDim2 CHILD OF
idDim2H1 )
ATTRIBUTE idDim2 DETERMINES (AtDim2)
ATTRIBUTE idDim2H1 DETERMINES (AtDim2H1);
```

Puisque la table_Dim3 ne contient qu'un seul niveau on n'a pas besoin d'introduire dans la dimension-2 une hiérarchie.

```
create DIMENSION Dim3
LEVEL idDim3 IS (Table_Dim3.idDim3)
ATTRIBUTE idDim3 DETERMINES (AtDim3) ;
```

Wondershare PDFelement

Explication des clauses utilisées dans la création des dimensions

level_clause
La clause level_clause définit un niveau dans la dimension. Un niveau définit les hiérarchies et les attributs de dimension.

level_table. level_column
Spécifiez les colonnes du niveau. Vous pouvez spécifier jusqu'à 32 colonnes. Les tables que vous spécifiez dans cette clause doivent déjà exister.

hierarchy_clause
La clause hierarchy_clause définit une hiérarchie linéaire de niveaux dans la dimension. Chaque hiérarchie forme une chaîne de relations parent-enfant entre les niveaux de la dimension. Les hiérarchies d'une dimension sont indépendantes les unes des autres. Ils peuvent, mais pas nécessairement, avoir des colonnes en commun. Chaque niveau de la dimension doit être spécifié au plus une fois dans cette clause, et chaque niveau doit déjà avoir été nommé dans la clause level_clause.

child_level
Spécifiez le nom d'un niveau qui a une relation n : 1 avec un niveau parent. Autrement dit, les level_columns de child_level ne peuvent pas être null et chaque valeur child_level détermine de manière unique la valeur du ***parent_level*** nommé suivant.
Si le niveau_table enfant est différent du niveau_table parent, vous devez spécifier une

Wondershare PDFelement

relation de jointure entre eux dans la dimension en utilisant la clause **JOIN KEY**.

parent_level
Spécifiez le nom d'un niveau parent qui une relation 1 : n.

Join Key pour spécifier une relation de jointure dans le cas où la table d'une dimension est normalisée c-a-d les colonnes source des niveaux d'une dimension proviennent de la même table.

clause_attribut
La clause attribute_clause vous permet de spécifier les colonnes qui sont uniquement déterminées par un niveau de hiérarchie. Les colonnes de niveau doivent toutes provenir de la même table que les colonnes_dépendantes. Les colonnes_dépendantes n'ont pas besoin d'être spécifiées dans la clause level_clause.

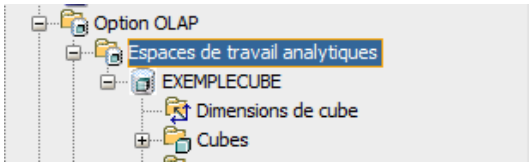
b- Création des dimensions avec SQL-developper (Option OLAP)

Wondershare PDFelement

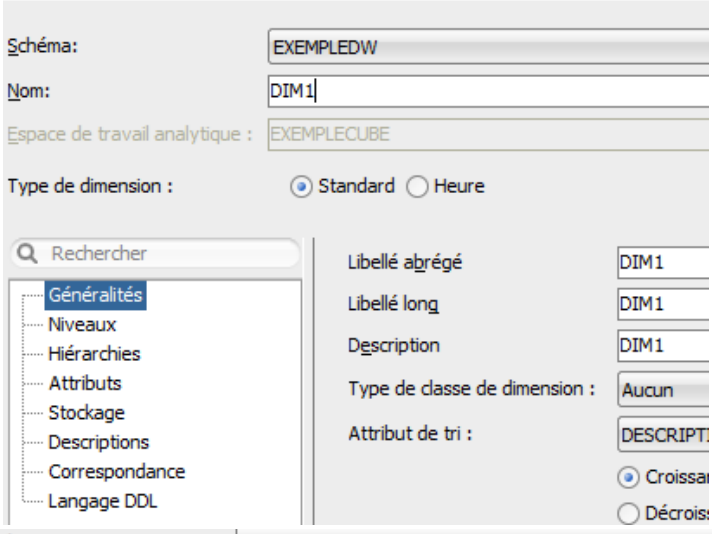
La création des dimensions en utilisant l'Option OLAP dans SQL developper se fait suivant les étapes suivantes :

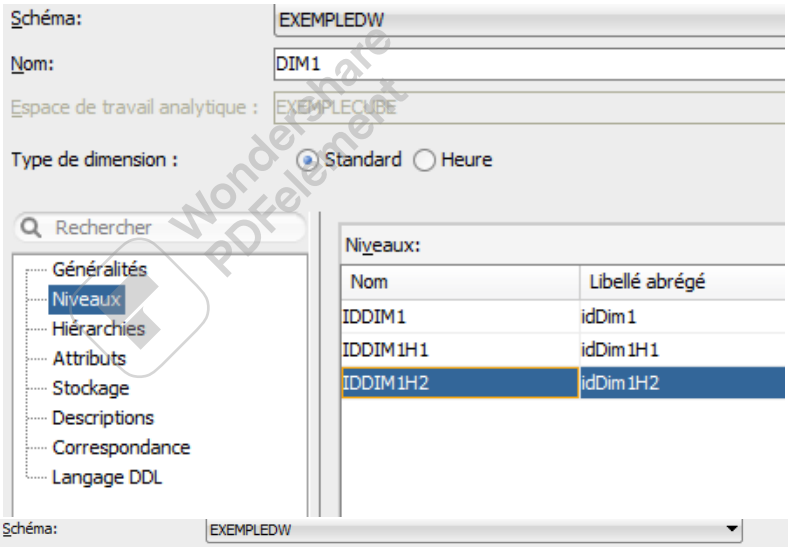
1- Création d'un nouvel espace de travail
2- Création d'une nouvelle dimension
3- Création des niveaux
4- Création des nouvelles hiérarchies puis spécifier niveaux disponibles pour chacune d'entre elles.
5- Créer les attributs puis appliquer les aux niveaux hiérarchiques
6- Dans stockage, choisir les contraintes de données hiérarchiques
7- Mappage des niveaux et des attributs avec les colonnes sources. Il faut préciser dans ce mappage les équi-jointures si les niveaux proviennent de plusieurs tables c-a-d le cas où les dimensions sont normalisées (modèle en flocon). Pour cela il faut mettre en correspondance les objets 'nonduniveau'. Parent avec le niveau père de la même table. Dans note modèle pris comme exemple on a mis en correspondance : IDDIM1H.PARENT avec TABLE_DIM1HA.IDDIM1H2
8- Validation de la création de la dimension
9- Construction de la dimension en chargeant les données dans la dimension. Une vue correspondant à la dimension sera créée.

Ci-dessous les images des captures d'ecran suivant l'ordre des étapes de création d'un cube de données mentionné ci-dessus.

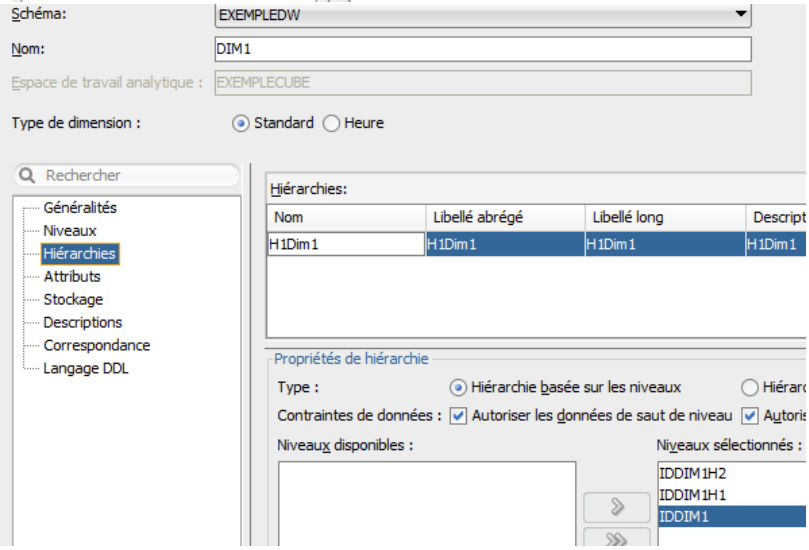


Wondershare PDFelement

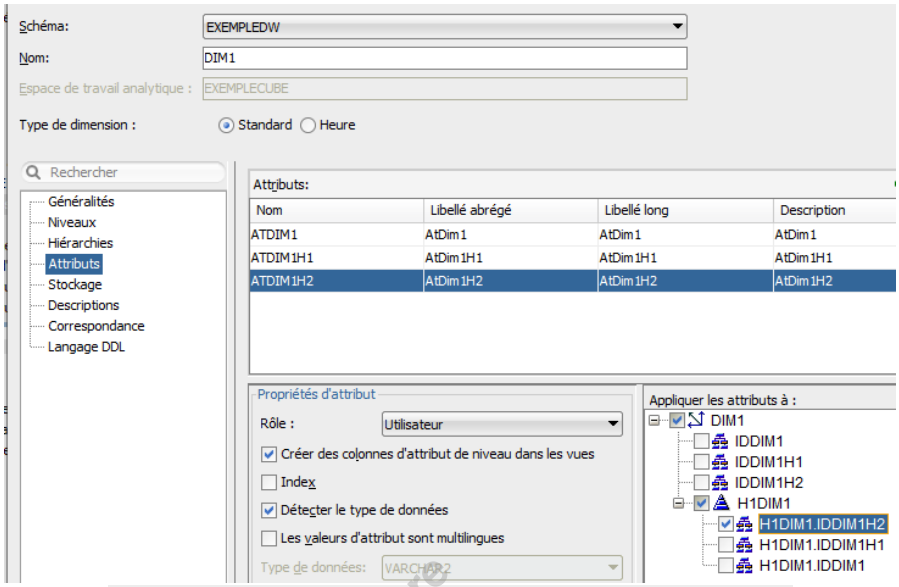


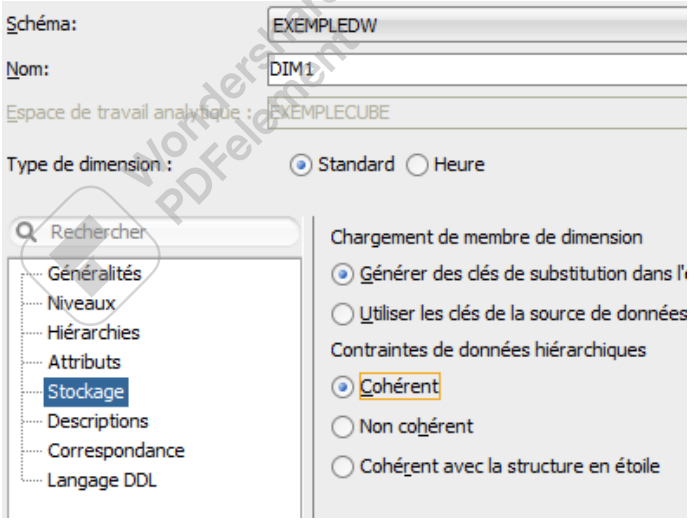


Wondershare PDFelement



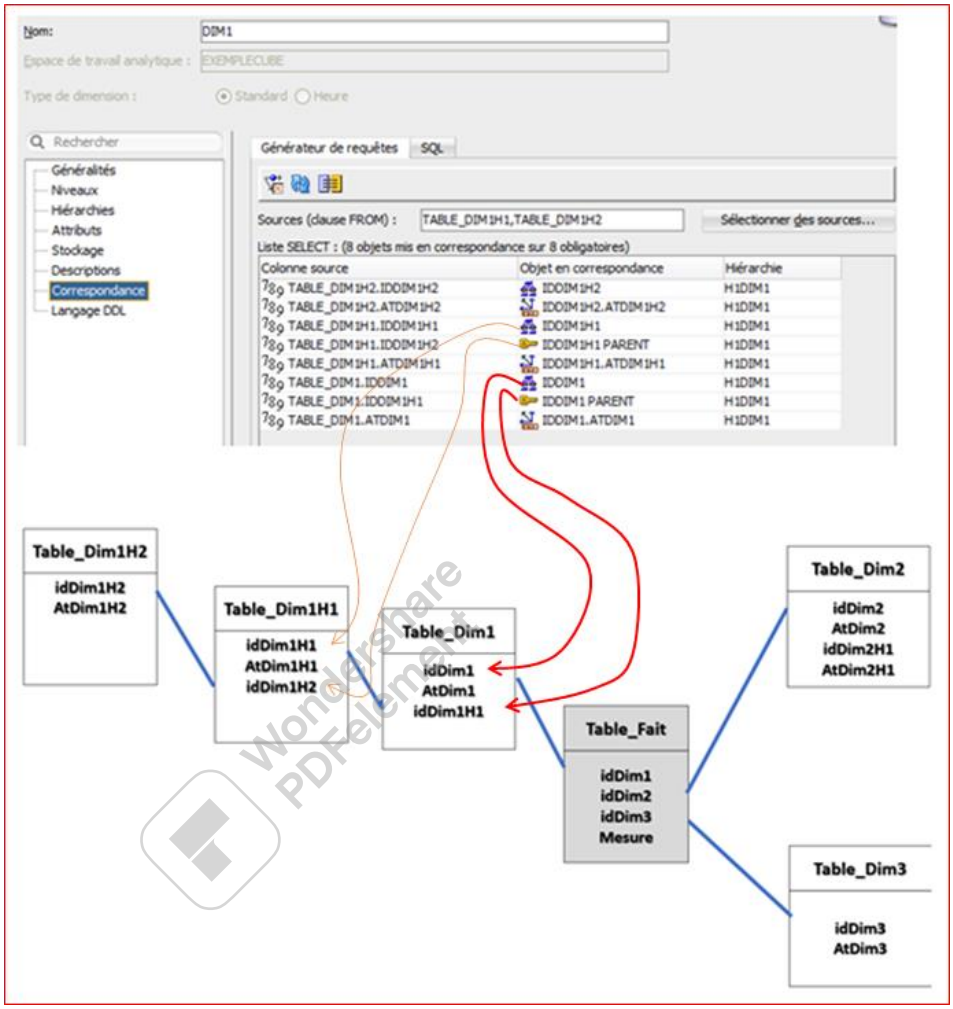
Wondershare PDFelement



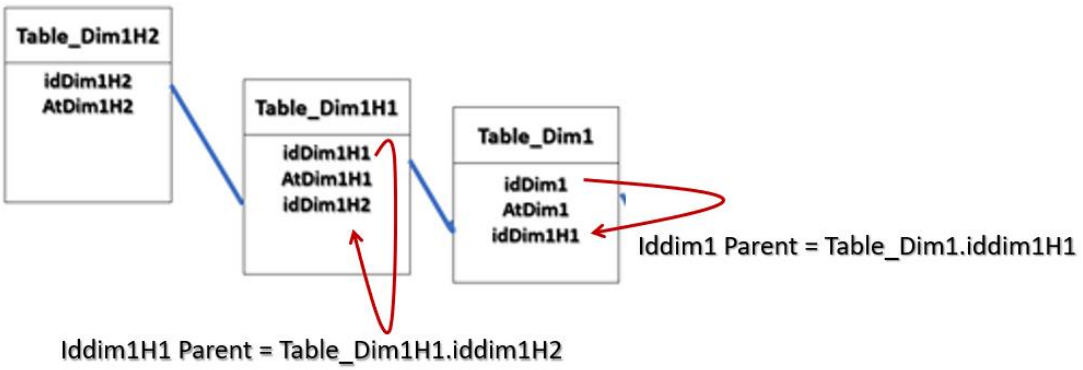


Wondershare PDFelement

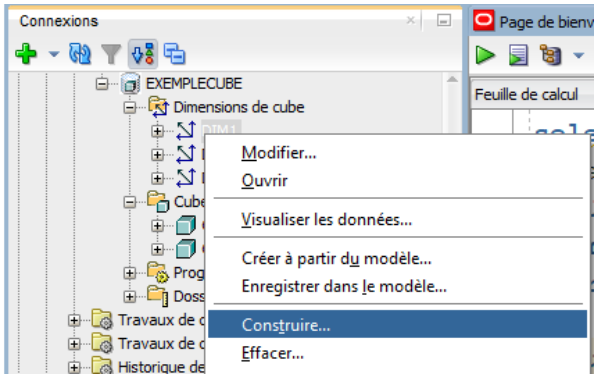
Wondershare PDFelement

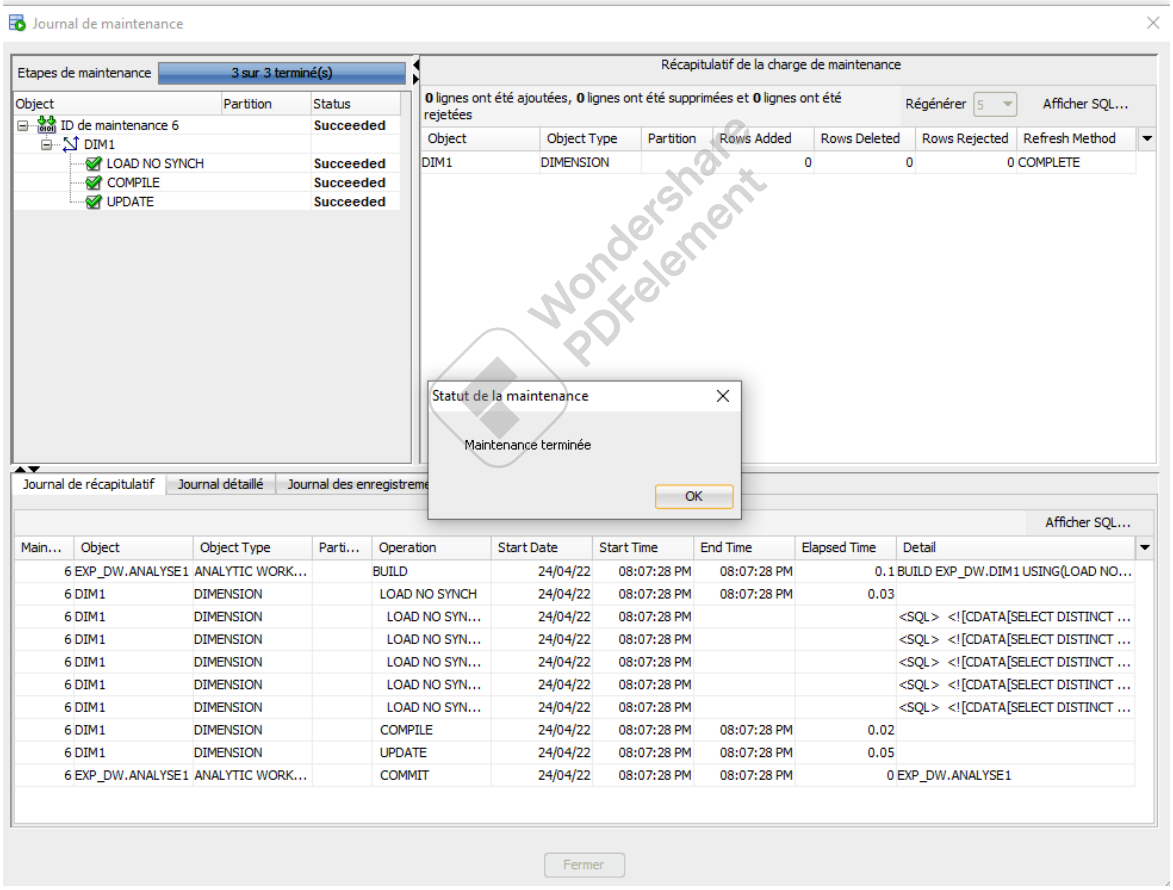


Dans le cas des dimensions normalisées, on doit préciser dans le mappage l'attribut « parrent » de chaque clé primaire des tables hiérarchiques. Ces attributs « parrents » sont les clés étrangères qui font références aux clés primaires des tables de hiérarchies supérieures. Dans notre exemple le mappage des « parrent key » est donné par la figure ci-dessous.



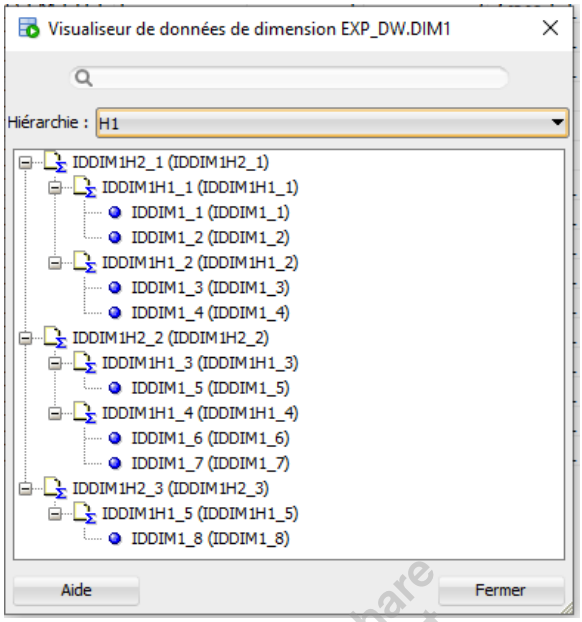
Wondershare PDFelement





La visualisation des données des dimensions permet de voir l'arborescence des différents niveaux d'une hiérarchie.

Wondershare PDFelement



La création des chaque dimensions d'une certaine hiérarchie généré une vue dont la structure est :

Pour la vue créer pour la hiérarchie H1 de la dimension-1 :

DIM_key	La clé des table dimension
LEVEL name	Le nom des niveaux
IdDim1	Identifiant du niveau dim1
IdDim1H1	Identifiant du niveau dim1H1
IdDim1H2	Identifiant du niveau dim1H2
ATDim1	Attribut lié au niveau dim1
ATDim1H1	Attribut lié au niveau dim1H1
ATDim1H2	Attribut lié au niveau dim1H2
DEPTH	La profondeur de chaque niveau hiérarchique

Wondershare PDFelement

Wondershare PDFelement

de Bienvenue | DIM1_H1_VIEW

Données | Droits | Dépendances | Détails | Déclencheurs | SQL | Erreurs

Trier... | Filtre : | Actions

DIM_KEY	LEVEL_NAME	ME...	DI...	HIE...	ATDIM1H2	ID...	ATDIM1H1	IDDIM1H1_...	ATDIM1	IDDIM1_ATDIM1	PARENT	DEPTH
IDDIM1H1 1	IDDIM1H1	L	1	2	(null)	a1	aa1	aa1	(null)	(null)	IDDIM1H2 1	1
IDDIM1H1 2	IDDIM1H1	L	2	5	(null)	a1	bb1	bb1	(null)	(null)	IDDIM1H2 1	1
IDDIM1H1 3	IDDIM1H1	L	3	9	(null)	b1	cc1	cc1	(null)	(null)	IDDIM1H2 2	1
IDDIM1H2 1	IDDIM1H2	L	6	1	a1	a1	(null)	(null)	(null)	(null)	(null)	0
IDDIM1H2 2	IDDIM1H2	L	7	8	b1	b1	(null)	(null)	(null)	(null)	(null)	0
IDDIM1H2 3	IDDIM1H2	L	8	14	c1	c1	(null)	(null)	(null)	(null)	(null)	0
IDDIM1 1	IDDIM1	L	9	3	(null)	a1	(null)	aa1	aaa1	aaa1	IDDIM1H1 1	2
IDDIM1 2	IDDIM1	L	10	4	(null)	a1	(null)	aa1	bbb1	bbb1	IDDIM1H1 1	2
IDDIM1 3	IDDIM1	L	11	6	(null)	a1	(null)	bb1	ccc1	ccc1	IDDIM1H1 2	2
IDDIM1H1 4	IDDIM1H1	L	4	11	(null)	b1	dd1	dd1	(null)	(null)	IDDIM1H2 2	1
IDDIM1H1 5	IDDIM1H1	L	5	15	(null)	c1	ff1	ff1	(null)	(null)	IDDIM1H2 3	1
IDDIM1 4	IDDIM1	L	12	7	(null)	a1	(null)	bb1	ddd1	ddd1	IDDIM1H1 2	2
IDDIM1 5	IDDIM1	L	13	10	(null)	b1	(null)	cc1	eee1	eee1	IDDIM1H1 3	2
IDDIM1 6	IDDIM1	L	14	12	(null)	b1	(null)	dd1	fff1	fff1	IDDIM1H1 4	2
IDDIM1 7	IDDIM1	L	15	13	(null)	b1	(null)	dd1	ggg1	ggg1	IDDIM1H1 4	2
IDDIM1 8	IDDIM1	L	16	16	(null)	c1	(null)	ff1	hhh1	hhh1	IDDIM1H1 5	2

VII-1-2 Création des cubes de données :

a- Syntaxe SQL de création des cubes de données

Il s'agit de l'instruction SELECT logique qui sera générée pour charger les données dans le cube CUBE1.

```
select
  --- Toutes les dimensions ont été mises en correspondance
  TABLE_FAIT.IDDIM1, -- Into DIM1.IDDIM1
  TABLE_FAIT.IDDIM2, -- Into DIM2.IDDIM2
  TABLE_FAIT.IDDIM3, -- Into DIM3.IDDIM3
  --- Tous les indicateurs ont été mis en correspondance
  SUM(TABLE_FAIT.MESURE) as MEASURE1
from TABLE_FAIT
group by TABLE_FAIT.IDDIM1, TABLE_FAIT.IDDIM2,
TABLE_FAIT.IDDIM3
```

Wondershare PDFelement

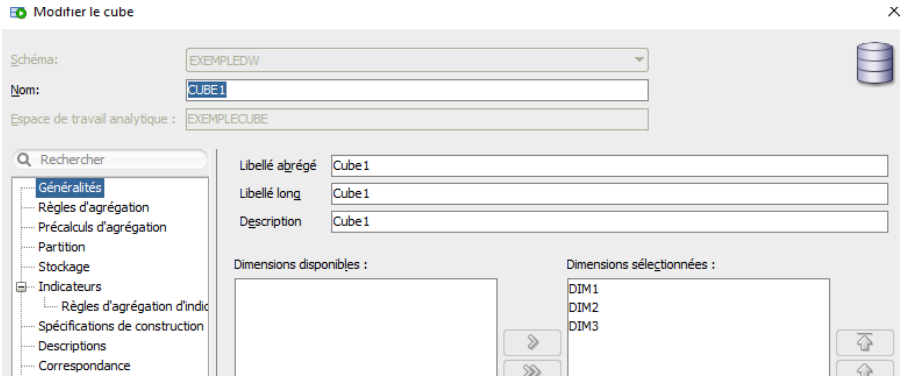
b- Création des cubes de données avec l'Option OLAP

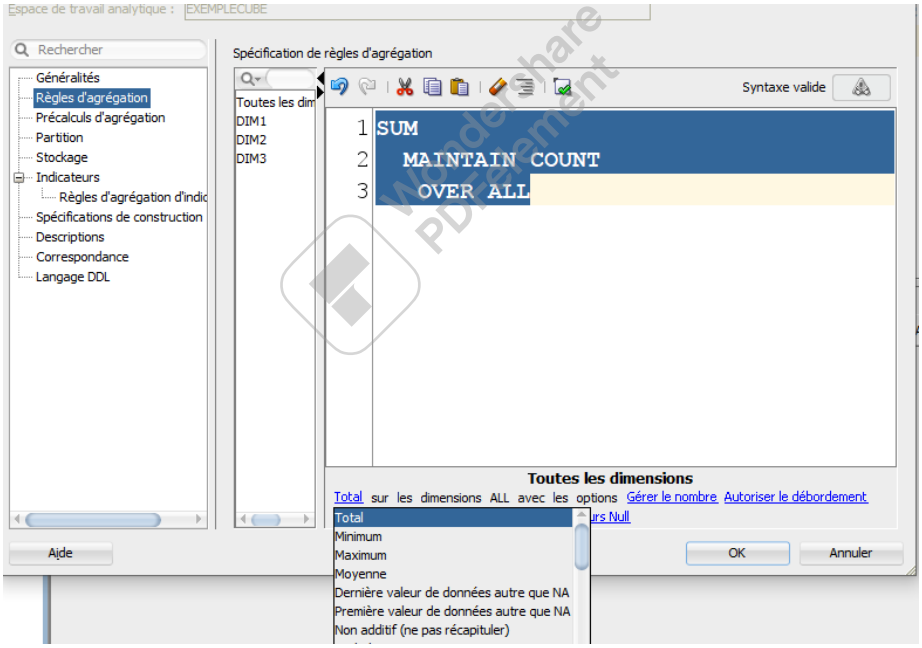
La création des Cubes en utilisant l'Option OLAP dans SQL developper se fait suivant les étapes suivantes :

1- Nommer le cube puis insérer les dimensions demandées pour analyser les mesures
2- Spécifier les règles d'agrégation en précisant l'opération de synthèse (SUM, AVG, MIN, ….)
3- Préciser la correspondance (Mappage) entre les colonnes source et les objets du cube
4- Validation la création du cube
5- Construction du cube en chargeant les données renvoyées par la requête de sélection. Une vue correspondant au cube de données sera créée.

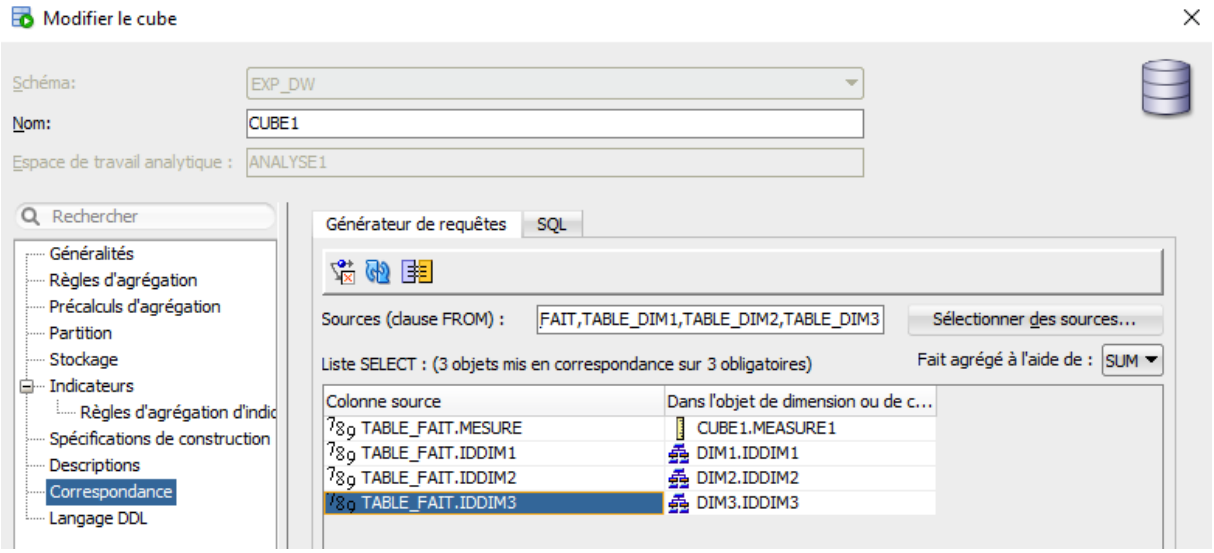
Wondershare PDFelement

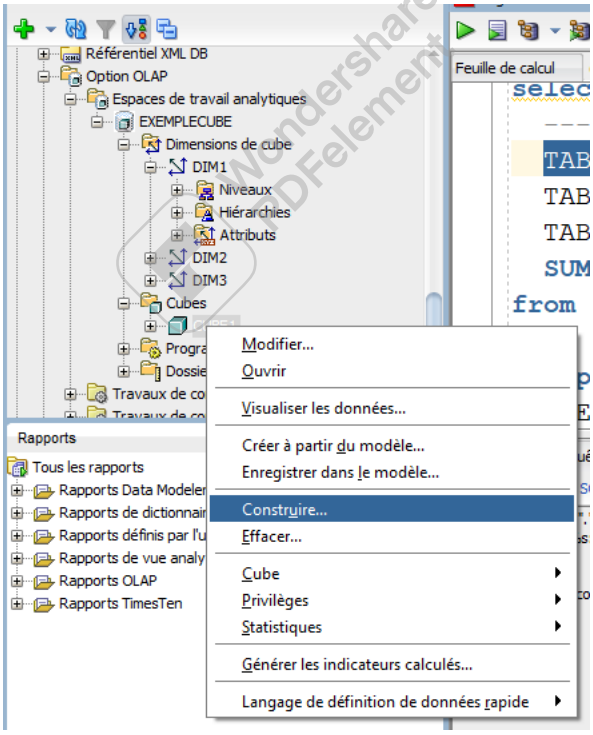
Ci-dessous les images des captures d'écran suivant l'ordre des étapes de création d'un cube de données mentionné ci-dessus.



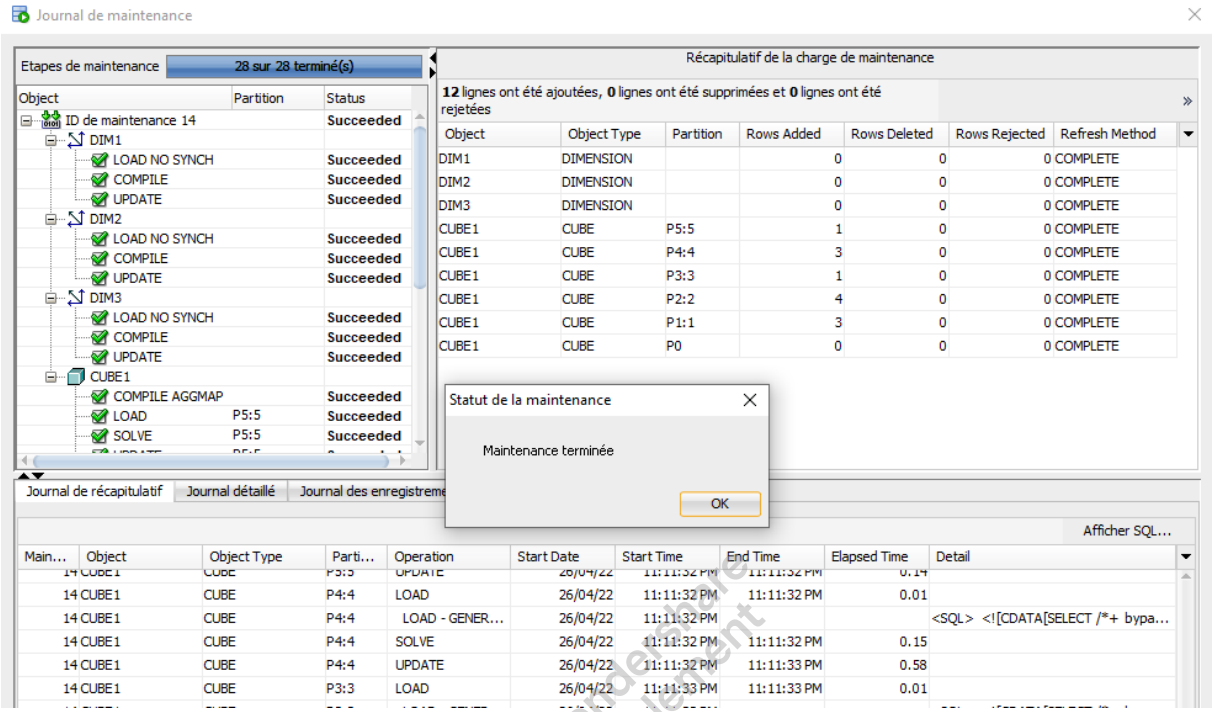


Wondershare PDFelement





Wondershare PDFelement

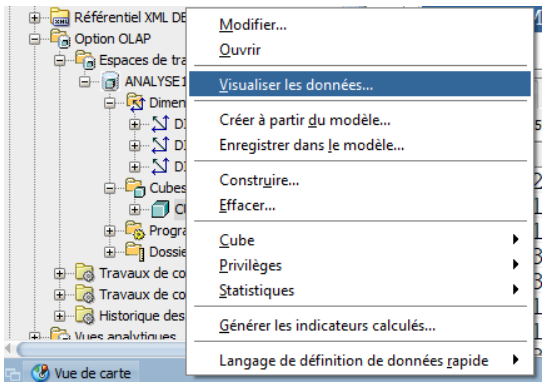


La création d'un cube de donnée génère la vue « nomCube_view » avec la structure suivante :

Attribut	Description
Mesure	L'objet de calcul d'analyse
Dim1	Contient les identifiants des différents niveaux hiérarchiques de la dimension1
Dim2	Contient les identifiants des différents niveaux hiérarchiques de la dimension2

Les données du cube de données correspondant au schéma multidimensionnel de la figure-8 peuvent être afficher de deux manières :

- Avec Option OLAP et avec le bouton droit sur le cube créé on choisit « visualiser les données ».



L'affichage du cube donnant la mesure suivant les 3 dimension :

Wondershare PDFelement

Pour la dim1 on affiche la hiérarchie la plus élevée, le niveau « IDDIM1H2 ».
La même chose avec la dim2, on affiche le niveau « IDDIM2H1 ».
Pour la dim3 le seul niveau est « IDDIM3 »

DIM1	DIM2	DIM3	MEASURE1
IDDIM1H2 1	IDDIM2H1 1	IDDIM3 1	1279,3
IDDIM1H2 1	IDDIM2H1 1	IDDIM3 2	548,2
IDDIM1H2 1	IDDIM2H1 2	IDDIM3 2	745,6
IDDIM1H2 1	IDDIM2H1 2	IDDIM3 3	8741,3
IDDIM1H2 2	IDDIM2H1 2	IDDIM3 3	785
IDDIM1H2 2	IDDIM2H1 2	IDDIM3 4	301,7
IDDIM1H2 3	IDDIM2H1 2	IDDIM3 5	174,7

Si on ajoute pour la dim1 le niveau hiérarchique IDDIM1H1 puis le niveau hiérarchique IDDIM1. On obtient

DIM1	DIM2	DIM3	MEASURE1
IDDIM1H2 1	IDDIM2H1 1	IDDIM3 1	1279,3
IDDIM1H2 1	IDDIM2H1 1	IDDIM3 2	548,2
IDDIM1H2 1	IDDIM2H1 2	IDDIM3 2	745,6
IDDIM1H2 1	IDDIM2H1 2	IDDIM3 3	8741,3
IDDIM1H1 1	IDDIM2H1 1	IDDIM3 1	1132,3
IDDIM1 1	IDDIM2H1 1	IDDIM3 1	236,2
IDDIM1 2	IDDIM2H1 1	IDDIM3 1	896,1
IDDIM1H1 2	IDDIM2H1 1	IDDIM3 1	147
IDDIM1H1 2	IDDIM2H1 1	IDDIM3 2	548,2
IDDIM1H1 2	IDDIM2H1 2	IDDIM3 2	745,6
IDDIM1H1 2	IDDIM2H1 2	IDDIM3 3	8741,3
IDDIM1 3	IDDIM2H1 1	IDDIM3 1	147
IDDIM1 3	IDDIM2H1 1	IDDIM3 2	548,2
IDDIM1 3	IDDIM2H1 2	IDDIM3 2	745,6
IDDIM1 4	IDDIM2H1 2	IDDIM3 3	8741,3

Wondershare PDFelement

- On peut afficher les données de l'hypercube on interrogant les vues générées avec des re quêtes SQL

Exemple : Pour afficher le cube suivant les 3 dimensions : « IDDIM1H2 », « IDDIM1H1 » et « IDDIM1 », on écrit :

```
select DIM1_H1DIM1_VIEW.DIM_KEY as "DIM1",
      DIM2_H1DIM2_VIEW.DIM_KEY as "DIM2",
      DIM3_H1DIM3_VIEW.DIM_KEY as "DIM3",
      round(MEASURE1,2) as "MEASURE1"
from  CUBE1_VIEW, DIM1_H1DIM1_VIEW,
     DIM2_H1DIM2_VIEW,
     DIM3_H1DIM3_VIEW
where CUBE1_VIEW.DIM1 = DIM1_H1DIM1_VIEW.DIM_KEY and
```

Wondershare PDFelement

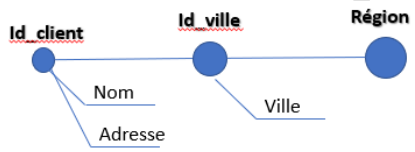

```
CUBE1_VIEW.DIM2 = DIM2_H1DIM2_VIEW.DIM_KEY and
CUBE1_VIEW.DIM3 = DIM3_H1DIM3_VIEW.DIM_KEY and
(DIM1_H1DIM1_VIEW.LEVEL_NAME in ('IDDIM1H2')) and
(DIM2_H1DIM2_VIEW.LEVEL_NAME in ('IDDIM2H1')) and
(DIM3_H1DIM3_VIEW.LEVEL_NAME in ('IDDIM3'));
```

VII-3 Exemple d'Implémentation de CUBES dans Oracle à l'aide de l'SQL

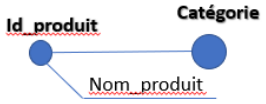
Nous somme intéressé par l'implémentation des cubes de données qui vont nous permettre d'analyser les ventes de produits. Par exemple : Quel est catégorie de produits a réalisé la meilleure vente sur une période donnée, dans une ville précise ?

La figure-7 illustre la création d'un cube de donnée à partie d'un datawarehouse modélisé dans un schéma en étoile. Les cubes obtenus représentent une analyse des ventes suivant trois dimensions : Dim_client, Dim_produit et Dim_Date. Chaque dimension est composée d'un ensemble de niveaux hiérarchiques qui déterminent le degré de granulation dans chaque dimension. Chaque niveau peut être décrit par des attributs. Ci-dessous les schémas des niveaux hiérarchiques de chaque dimension :

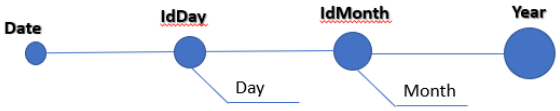
- Pour la dimension Dim_Client :



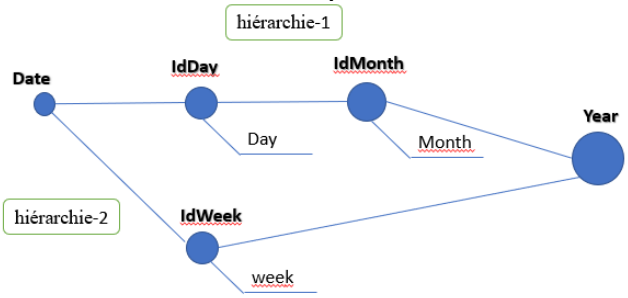
- Pour la dimension Dim_produit :



- Pour la dimension Dim_Date :



On peut utiliser une hiérarchie composée dans la dimension date qui va nous permettre de calculer les ventes de chaque semaine de l'année.



Wondershare PDFelement

Wondershare PDFelement

Exemple d'hypercube OLAP : Cube des Ventes
Produit × Ville × Mois → Ventes
Produit × Région × Année → Ventes
Catégorie × Ville × Mois → Ventes

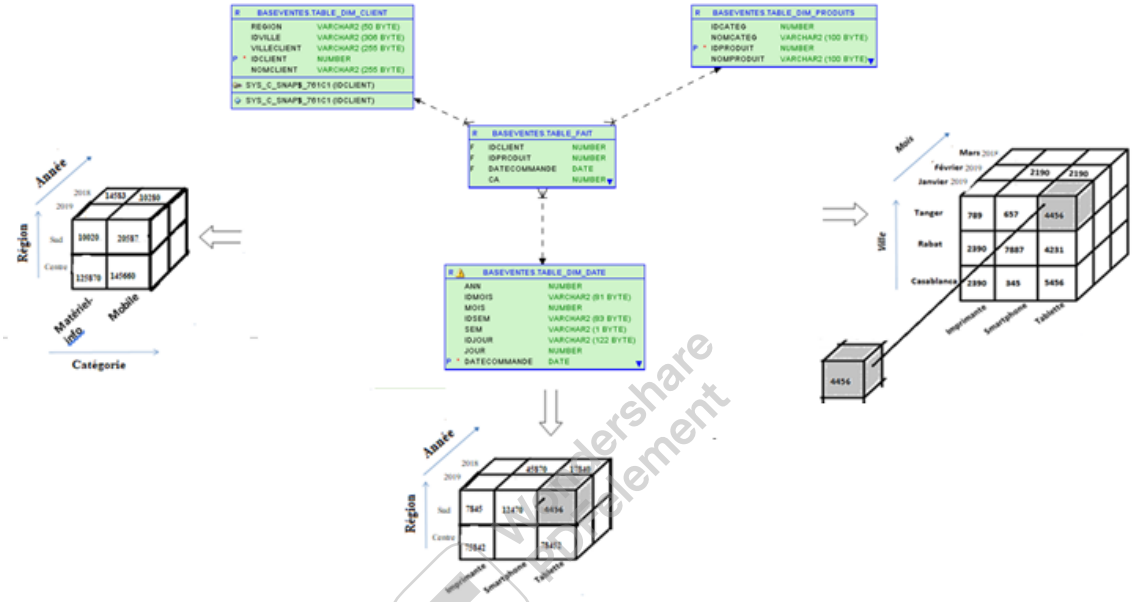


Figure-7 : HyperCube de données représentant les ventes suivant trois dimensions

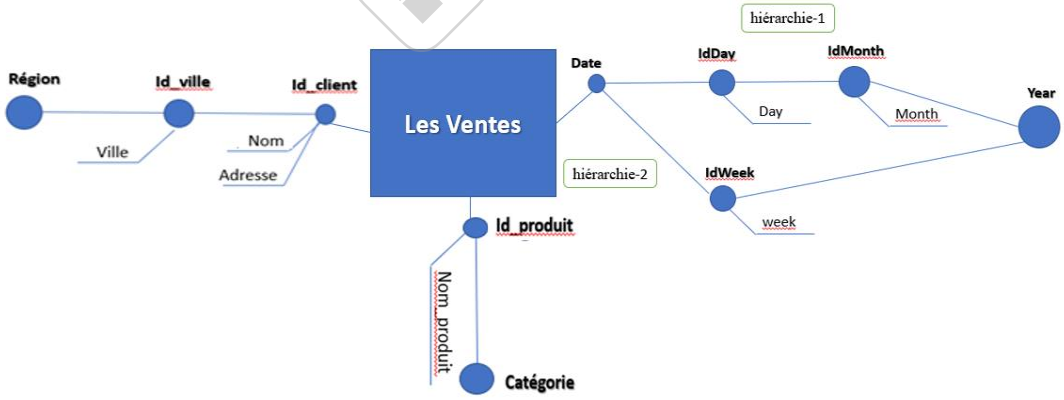


Figure-8 : Schéma multidimensionnelle d'analyse des ventes

a- Implémentation de l'entrepôt de données

- Création de la table dimension client

```
create materialized view Table_Dim_client
REFRESH COMPLETE on DEMAND as
select region, villeclient||'_'||region as idville, villeclient,
idclient, nomclient
from clients;
```

Wondershare PDFelement

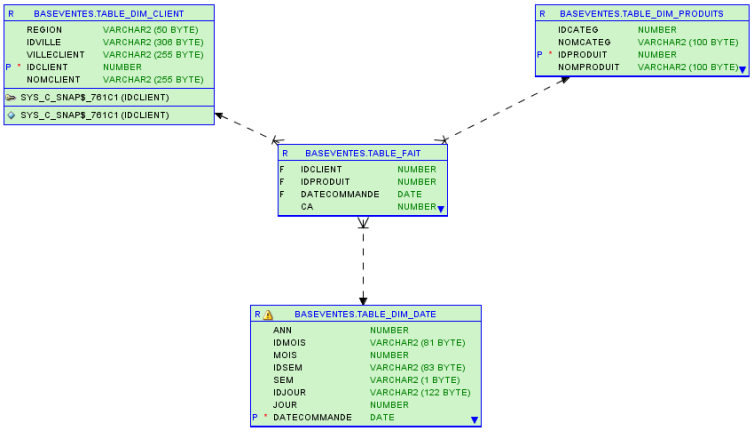
- Création de la table dimension Date

```
create MATERIALIZED VIEW Table_Dim_Date
REFRESH COMPLETE on DEMAND as
select extract(year from datecommande) as Ann,
extract(month from datecommande)||' '||
extract(year from datecommande)as idmois,
extract(month from datecommande) as Mois,
TO_CHAR(datecommande, 'd')||' '||extract(month from datecommande)
||' '||extract(year from datecommande)as idsem,
TO_CHAR(datecommande, 'd')  as sem,
extract(day from datecommande)||' '||extract(month from datecommande)
||' '||extract(year from datecommande) as idjour,
extract(day from datecommande) as Jour,
datecommande
from commandes;
```

Wondershare PDFelement

- Création de la table dimension Produit

```
create MATERIALIZED VIEW Table_Dim_Produits
REFRESH COMPLETE on DEMAND as
select categories.idcateg, nomcateg, idproduit, Nomproduit
from categories, produits
where categories.idcateg=produits.idcateg
```



b- Construction des hypercube de Données

Wondershare PDFelement

Cube-1: Analyse par dimension Date – hiérarchie-1 (date → day →month→year) :

DIM_DATE	MEASURE1
YEAR 2021	226636,39
YEAR 2022	350073,18
YEAR 2023	123660,39

DIM_DATE	MEASURE1
YEAR 2021	226636,39
IDMONTH 10 2021	13113,98
IDMONTH 11 2021	17306,55
IDMONTH 12 2021	14480,17
IDMONTH 5 2021	17267,99
IDMONTH 6 2021	42530,97
IDMONTH 7 2021	32363,31
IDMONTH 8 2021	56021,08
IDMONTH 9 2021	33552,33
YEAR 2022	350073,18
IDMONTH 1 2022	27210,31
IDMONTH 10 2022	16516,97
IDMONTH 11 2022	22342,96
IDMONTH 12 2022	44699,49
IDMONTH 2 2022	26560,32
IDMONTH 3 2022	13213,54
IDMONTH 4 2022	31881,33

DIM_DATE	MEASURE1
YEAR 2021	226636,39
IDMONTH 10 2021	13113,98
IDDAY 19 10 2021	3322,98
IDDAY 24 10 2021	1861,21
IDDAY 3 10 2021	7725,17
IDDAY 5 10 2021	204,61
IDMONTH 11 2021	17306,55
IDDAY 10 11 2021	5897,84
IDDAY 13 11 2021	4292,13
IDDAY 8 11 2021	7116,58
IDMONTH 12 2021	14480,17
IDDAY 16 12 2021	3443,01
IDDAY 23 12 2021	10193,17
IDDAY 31 12 2021	843,99
IDMONTH 5 2021	17267,99
IDDAY 12 5 2021	2295,67
IDDAY 22 5 2021	4180,54
IDDAY 29 5 2021	8479,11
IDDAY 31 5 2021	2312,67
IDMONTH 6 2021	42530,97
IDDAY 1 6 2021	6392,44
IDDAY 13 6 2021	2443,76
IDDAY 14 6 2021	2812,65

Cube-1: Analyse par dimension Date – hiérarchie-2 (date → week → year) :

DIM_DATE	MEASURE1
YEAR 2021	226636,39
YEAR 2022	350073,18
YEAR 2023	123660,39

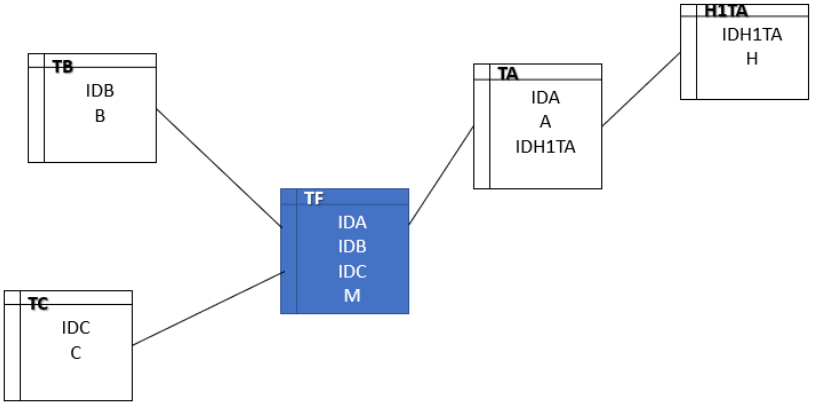
DIM_DATE	MEASURE1
YEAR 2021	226636,39
IDWEEK 1 2021	22538,28
IDWEEK 2 2021	23772,97
IDWEEK 3 2021	31132,76
IDWEEK 4 2021	43969,42
IDWEEK 5 2021	6365,08
IDWEEK 6 2021	63265,82
IDWEEK 7 2021	35592,07
YEAR 2022	350073,18
IDWEEK 1 2022	69982,04
IDWEEK 2 2022	39068,78
IDWEEK 3 2022	59180,4
IDWEEK 4 2022	22881,27
IDWEEK 5 2022	56700,88
IDWEEK 6 2022	57513,82
IDWEEK 7 2022	44746
YEAR 2023	123660,39
IDWEEK 1 2023	33123,05
IDWEEK 2 2023	10121,97
IDWEEK 3 2023	7720,24
IDWEEK 4 2023	12675,13

Wondershare PDFelement

VII-2 Opération sur les hypercubes de données

Plusieurs opérations peuvent être effectuées sur les cubes de données. On distincts les opérateurs sur les cubes (Opération de sélection : Slicing et Dicing) et les opérateurs sur les dimensions qui permettent de changer la granularité des dimensions (Roll up et Drill Down). Nous illustrons ces différentes opérations sur le modèle suivant :

Wondershare PDFelement



La fonction d'agrégation GROUP BY est utilisé pour créer le cube de données :

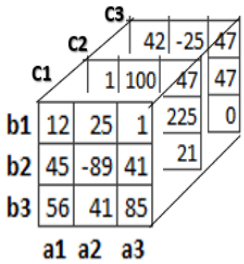
```
select A, B, C, sum(M)
from TA, TB, TC, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
AND TC.IDC=TF.IDC
group by A, B, C
order by A, B, C;
```

A	B	C	SUM(M)
a1	b1	c1	12
a1	b1	c2	1
a1	b1	c3	42
a1	b2	c1	45
a1	b2	c2	0
a1	b2	c3	-12
a1	b3	c1	56
a1	b3	c2	54
a1	b3	c3	71
a2	b1	c1	25
a2	b1	c2	100
a2	b1	c3	-25
a2	b2	c1	-89
a2	b2	c2	22
a2	b2	c3	32
a2	b3	c1	41
a2	b3	c2	25
a2	b3	c3	78
a3	b1	c1	1

C1	a1	a2	a3
b1	12	25	1
b2	45	-89	41
b3	56	41	85

C2	a1	a2	a3
b1	1	100	47
b2	0	22	225
b3	54	25	21

C3	a1	a2	a3
b1	42	-25	47
b2	-12	32	47
b3	71	78	0



VII-2-1 Opération de sélection (Slicing et Dicing)

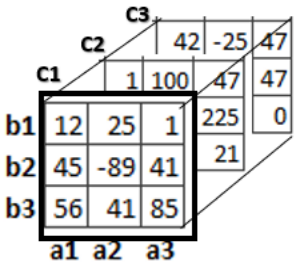
Slicing: Sélection des surfaces du cube par des prédicats selon une dimension

Wondershare PDFelement


```
select A, B, sum(M)
from TA, TB, TC, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
AND TC.IDC=TF.IDC
group by A, B, C
having C ='c1'
order by A, B;
```

A	B	SUM(M)
a1	b1	12
a1	b2	45
a1	b3	56
a2	b1	25
a2	b2	-89
a2	b3	41
a3	b1	1
a3	b2	41
a3	b3	85

C1	a1	a2	a3
b1	12	25	1
b2	45	-89	41
b3	56	41	85



Dicing: extraction d'un sous-cube ou Sélection des sous-cubes du cube par des prédicats selon plusieurs dimensions

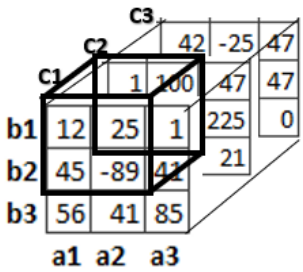
```
select A, B, sum(M)
from TA, TB, TC, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
AND TC.IDC=TF.IDC
group by A, B, C
having C !='c1'
AND A != 'a3'
AND B!='b3'
order by A, B;
```

A	B	SUM(M)
a1	b1	1
a1	b1	42
a1	b2	-12
a1	b2	0
a2	b1	100
a2	b1	-25
a2	b2	32
a2	b2	22

Wondershare PDFelement

C1	a1	a2
b1	12	25
b2	45	-89

C2	a1	a2
b1	1	100
b2	0	22



VII-2-2 Opérations de changement de granularité (Roll-up et Drill-Down)

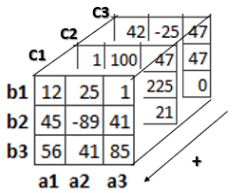
Ces opérations agissent sur la granularité d'observation des données caractérisent la hiérarchie de navigation entre les différents niveaux.

Roll-up: consiste à représenter les données du cube à un niveau de granularité supérieur conformément à la hiérarchie définie sur la dimension. Utilisation dans ce cas des fonctions d'agrégation (somme, moyenne, etc)

Wondershare PDFelement

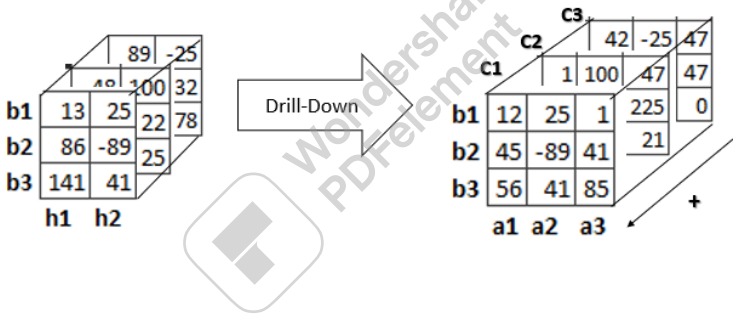

```
select A, B, sum(M)
from TA, TB, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
group by A, B
order by A, B;
```

A	B	SUM(M)
a1	b1	55
a1	b2	33
a1	b3	181
a2	b1	100
a2	b2	-35
a2	b3	144
a3	b1	95
a3	b2	313
a3	b3	106



	a1	a2	a3
b1	55	100	95
b2	33	-35	313
b3	181	144	106

Drill-down: consiste à représenter les données du cube à un niveau de granularité de niveau inférieur, donc sous une forme plus détaillée. Dans notre exemple change la granularité du niveau H au niveau A



Wondershare PDFelement

VII-3 Les fonction d'agrégation dans les DWs

Outre les résultats d'agrégation réguliers attendus de la clause GROUP BY, il existe des extensions qui permettent de générer des sous-totaux (CUBE et ROLLUP)

VII-3-1 ROLLUP

Outre les résultats d'agrégation réguliers attendus de la clause GROUP BY, l'extension ROLLUP génère des sous-totaux de groupe de droite à gauche et un total général.

Wondershare PDFelement


```
select A,B, COUNT(*)
from TA, TB, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
group by ROLLUP(A, B);
```

A	B	COUNT(*)
a1	b1	6
a1	b2	5
a1	b3	7
a1	(null)	18
a2	b1	6
a2	b2	7
a2	b3	7
a2	(null)	20
a3	b1	7
a3	b2	9
a3	b3	8
a3	(null)	24
(null)	(null)	62

V-4-1 CUBE

Outre les sous-totaux générés par l'extension ROLLUP, l'extension CUBE génère des sous-totaux pour toutes les combinaisons des dimensions spécifiées.

Wondershare PDFelement

```
select A,B, COUNT(*)
from TA, TB, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
group by cube(A, B);
```

A	B	COUNT(*)
(null)	(null)	62
(null)	b1	19
(null)	b2	21
(null)	b3	22
a1	(null)	18
a1	b1	6
a1	b2	5
a1	b3	7
a2	(null)	20
a2	b1	6
a2	b2	7
a2	b3	7
a3	(null)	24
a3	b1	7
a3	b2	9
a3	b3	8

V-4-1 GROUPING SETS

Outre les sous-totaux générés par l'extension CUBE, l'extension GROUPING SETS restreint l'affichage aux sous-totaux seulement.

Wondershare PDFelement


```
select A,B, COUNT(*)
from TA, TB, TC, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
group by GROUPING SETS(A, B);
```

A	B	COUNT(*)
a1	(null)	18
a3	(null)	24
a2	(null)	20
(null)	b2	21
(null)	b1	19
(null)	b3	22

V-4-1 Fonction Decode

avec la fonction DECODE() on peut afficher une chaine de caractère au lieu de « NULL ».

```
select decode(GROUPING(A),1,'toute les A', A) as A,
decode(GROUPING(B),1,'toute les B',B) as B, COUNT(*)
from TA, TB, TF
where TA.IDA=TF.IDA
AND TB.IDB=TF.IDB
group by cube(A, B);
```

A	B	COUNT(*)
toute les A	toute les B	62
toute les A	b1	19
toute les A	b2	21
toute les A	b3	22
a1	toute les B	18
a1	b1	6
a1	b2	5
a1	b3	7
a2	toute les B	20
a2	b1	6
a2	b2	7
a2	b3	7
a3	toute les B	24
a3	b1	7
a3	b2	9
a3	b3	8

Wondershare PDFelement

<u>Fonction DECODE()</u>

C'est fonction programmée dans la bibliothèque d'oracle qui permet de faire un traitement conditionnel sans passer par un langage supérieur (PL/SQL) :

Syntaxe :

- Pour un traitement conditionnel simple : *DECODE(X, C1, V, F)*

si X vérifie la Condition C1 Alors
renvoyer V
sinon
renvoyer F
Fin Si

Wondershare PDFelement

- Pour un traitement conditionnel multiple : *DECODE(X, C1, V1, C2, V2, C3, V3, ……)*

si X vérifie la Condition C1 Alors
renvoyer V1
    sinon Si X vérifie la Condition C2 Alors
    renvoyer V2
        sinon Si X vérifie la Condition C3 Alors
        renvoyer V3
        .....
        .....
Fin Si

Wondershare PDFelement